MINISTRY OF EDUCATION

Robotics

for Senior High Schools

TEACHER MANUAL

YEAR 3

NATIONAL COUNCIL FOR CURRICULUM & ASSESSMENT OF MINISTRY OF EDUCATION

MINISTRY OF EDUCATION

REPUBLIC OF GHANA

Robotics

For Senior High Schools

Teacher Manual

Year Three

NaCCA
NATIONAL COUNCIL FOR
CURRICULUM & ASSESSMENT
OF MINISTRY OF EDUCATION

ROBOTICS TEACHER MANUAL

Enquiries and comments on this manual should be addressed to:

The Director-General

National Council for Curriculum and Assessment (NaCCA)

Ministry of Education

P. O. Box CT PMB 77 Cantonments Accra

Telephone: 0302909071, 0302909862

Email: info@nacca.gov.gh

Website: www.nacca.gov.gh

Contents

List of Figures & Tables

Introduction

The National Council for Curriculum and Assessment (NaCCA) has developed the third and final set of curriculum materials to support the implementation of the new Senior High School curriculum.

This Year Three Teacher Manual for covers all aspects of the content, pedagogy, teaching and learning resources and assessment required to effectively support learners in their final year of Senior High School. All the necessary information for the Student Assessment Portal has been included in the Year Three Teacher Manual to simplify the development of weekly learning plans, meaning that teachers have a single reference document.

The Year Three Teacher Manual has two main objectives:

- To support learners to consolidate and deepen their understanding of the content, pedagogy, and assessment across all three years of the new Senior High School curriculum in preparation for the West Africa Senior Secondary Certificate Examination (WASSCE).
- To equip teachers with the tools and resources needed to deliver revision-focused, assessment-led lessons that build on the knowledge learners have acquired from Year One to Year Three.

In this third year, teachers will support learners in preparing for the WASSCE. Each section of the Teacher Manual is therefore structured around the highest Depth of Knowledge (DoK) levels, allowing learners to build progressively on knowledge acquired from Year One to Year Three. For each week, learning tasks and pedagogy are built around the assessment, ensuring that classroom activities directly prepare learners for the demands of the examination. Two sets of sample examination papers are also provided; one administered by Week 12 as a first semester practice examination, and one by Week 20 as a mock examination, to gauge learners' competencies and identify areas requiring remediation.

To further support teachers in delivering Year Three, NaCCA, working with the Ghana Education Service, has developed a subject specific App for each subject.

The Year Three PLC sessions have been revised in response to findings from the 2025 annual evaluation survey. The survey confirmed that teachers have made strong progress in lesson planning, use of varied instructional methods, ICT integration and modes of assessment. To address the identified gap in explaining concepts clearly using examples familiar to learners, each PLC session now includes a dedicated Content Exploration segment. This ten-minute segment brings the subject group together to work through one challenging problem or concept from the week's Teacher Manual content, not to plan how to teach it, but to ensure that every teacher in the group fully understands it.

The PLC sessions in Year Three also place particular emphasis on connecting the week's content to equivalent topics taught in Years One and Two. Teachers are encouraged to note the recap weeks signposted throughout the Teacher Manual and to use the App to explore how content across all three years links together.

As in previous years, teachers are expected to integrate National Values, Gender Equality and Social Inclusion (GESI), Social and Emotional Learning (SEL), and 21st-century skills into their learning activities, assessment tasks, and rubrics and mark schemes. The subject-specific App will support teachers in doing this throughout the year.

Ghana Education Service acknowledges the dedication of the teachers who have implemented the new curriculum across Years One and Two and recognises that Year Three represents both the culmination of that effort and a critical moment for learners. The Year Three Teacher Manual and its accompanying resources have been developed to ensure that every learner, regardless of background, ability, or school context, is fully prepared to demonstrate what they know and can do in the WASSCE and to progress confidently to further studies, the world of work, and adult life.

Acknowledgements

Special thanks to Professor Samuel Ofori Bekoe, Director-General of the National Council for Curriculum and Assessment (NaCCA) and all who contributed to the successful writing of the Teacher Manuals for the new Senior High School (SHS) curriculum.

The writing team was made up of the following members:

National Council for Curriculum and Assessment (NaCCA)	
Name of Staff	**Designation**
Reginald Quartey	Ag. Director, Curriculum Development Directorate
Anita Cordei Collison	Ag. Director, Standards, Assessment and Quality Assurance Directorate
Dr. Mercy Nyamekye	Ag. Director, Research, Planning, Monitoring, and Evaluation
Joana Adua Vanderpuije	Ag. Director, Instructional Resources Directorate
Rebecca Abu Gariba	Ag. Director, Corporate Affairs
Nii Boye Tagoe	Senior Curriculum Development Officer
Juliet Owusu-Ansah	Principal Curriculum Development Officer
Samuel Owusu Ansah	Principal Curriculum Development Officer
Godwin Mawunyo Kofi Senanu	Assistant Curriculum Development Officer
Stephen Acquah	Principal Curriculum Development Officer
Thomas Kumah Osei	Principal Curriculum Development Officer
Ayuuba Sullivan	Principal Curriculum Development Officer
Dr.Jephtar Adu-Mensah	Principal Standards, Assessment and Quality Assurance Officer (SAQA)
Richard Teye	Principal Standards, Assessment and Quality Assurance Officer (SAQA)
Nancy Asieduwaa Gyapong	Assistant Standards, Assessment and Quality Assurance Officer (SAQA)
Bridget Aku Anku	Principal, Standards, Assessment and Quality Assurance Officer (SAQA)
Francis Agbalenyo	Principal Research, Planning, Monitoring and Evaluation Officer
Joseph Barwuah	Senior Instructional Resource Officer
Samuel Amankwa Ogyampo	Senior Corporate Affairs Officer
Seth Nii Nartey	Corporate Affairs Officer

Writer	Name of Institution	Subject
Bernard Adobaw	West African Examinations Council	**Accounting**
Emmanuel Kodwo Arthur	Instittue of Charted Accountants, Ghana	
Benedicta Ama Yekua Etuaful	Ogyeedom SHTS	**Additional Mathematics**
Innocent Nunana Duncan	KNUST SHS	
Benedicta Carbilba Foli	Pope Johns SHS & Minor Seminary, Koforidua	**Agricultural Science**
David Esela Zigah	Achimota School	
Prof. Jakpasu Victor Kofi Afun	Kwame Nkrumah University of Science and Technology	
Eric Morgan Asante	St. Peter's Boys SHS	**Agriculture**
Prof. Frederick Adzitey	University for Development Studies, Tamale	
Akuffo Twumhene Frederick	Koforidua SHTS	**Applied Technology**
Emmanuel Korletey	Benso SHTS	
Firmin Anewuoh	Presbyterian College of Education, Akropong	
Isaac Buckman	Armed Forces SHTS	
Kunkyuuri Philip	Kumasi Senior High Technical School	
Lavoe Daniel Kwaku	Sokode SHTS	
Philip Turkson	St Joseph's SHTS, Bekwai	
Dr. Sherry Kwabla Amedorme	Univ. of Skills Training and Entrepreneurial Development	
Walter Banuenumah	Univ. of Skills Training and Entrepreneurial Development	
Wisdom Dzidzienyo Adzraku	Univ. of Skills Training and Entrepreneurial Development	
Abass Umar Mohammed	University of Ghana, Legon	**Arabic**
Mohammed Mahey Ibrahim	Tijjaniya Senior High School, Kumasi	
Dr. Ebenezer Acquah	University of Education, Winneba	**Art and Design Studio and Foundation**
Dr. Mantey Jectey-Nyarko	Kwame Nkrumah University of Science and Technology	
Kwame Opoku-Bonsu	Kwame Nkrumah University of Science and Technology	
Seyram kojo Adipah	GES- Ga East	
David Kofi Oppong	Kwame Nkrumah University of Science and Technology	**Aviation and Aerospace**
Obed Frimpong	Zipline International	

Jo Ann Naa Dei Neequaye	Nyakrom SHTS	**Biology**
Paul Beeton Damoah	Prempeh College, Kumasi	
Dr. Daniel Akwei Addo	Kwame Nkrumah University of Science and Technology	**Biomedical Science**
Dr. Mrs. Dorothy Araba Yakoba Agyapong	Kwame Nkrumah University of Science and Technology	
Avole Baba Ansbert	Bolga SHS	**Business Management**
Emmanuel Caesar Ayamba	Bolgatanga Technical University	
Faustina Graham	GES HQ	
Bismark Kwame Tunu	Opoku Ware School, Kumasi	**Chemistry**
Patrick Nsobila Awumbire	Bolgatanga S.H.S.	
Francis Normanyo	Mt Mary College of Education	**Christian Religious Studies & Islamic Religious Studies**
Francis Opoku	Valley View University	
Haruna Zagoon-Sayeed	University of Ghana	
Kabiru Soumana	GES HQ	
Richardson Addai-Mununkum	University of Education, Winneba	
Seth Tweneboah	University of Education, Winneba	
Jusinta Kwakyewaa (Rev.Sr.)	St Francis SHTS, Akim Oda	**Clothing and Textiles**
Rahimatu Yakubu	Potsin T.I Ahmadiyya SHS	
Dordoe Raphael Senyo	Ziavi SHTS	**Computing and ICT**
Gaddafi Abdul-salaam	Kwame Nkrumah University of Science and Technology	
Kwasi Abankwa Anokye	Ghana Education Service	
Gabriel Boafo	Kwabeng Anglican Senior High Technical School	**Design and Communication Technology**
Mensah Henry Angmor	Anglican Senior High School, Kumasi	
Phyllis Mensah	University of Skills Training and Entrepreneurial Development (	
Charlotte Kpogli	Ho Technical University	**Economics**
Salitsi Freeman Etornam	Anlo SHS, Anloga	
Daniel Kwesi Agbogbo	Kwabeng Anglican Senior High Tech School	**Engineering**
Dr Valentina Osei-Himah	Atebubu College of Education	

Benjamin Orrison Akrono	Islamic Girls' SHS	**English Language**
Esther Okaitsoe Armah	Mangoase SHS	
Kukua Andoh Robertson	Breman Asikuma SHS	
Perfect Quarshie	Mawuko Girls SHS, Ho	
Ama Achiaa – Afriyie	St. Louis SHS, Kumasi	**Food and Nutrition**
Lily-Versta Nyarko	Mancell Girls SHTS, Kumasi	
Osmanu Ibrahim	Mount Mary College of Education, Somanya	**French**
Mawufemor Kwame Agorgli	Akim Asafo SHS	
Comfort Korkor Sam	University for Development Studies, Tamale	**General Science**
Robert Arhin	S.D.A SHS, Akim Sekyere	
Ishmael Yaw Dadson	University of Education, Winneba	**Geography**
Prof Ebenezer Owusu -Sekyere	University for Development Studies, Tamale	
Raymond Nsiah-Asare	Methodist Girls' SHS, Mamfe	
Catherine Ekua Mensah	University of Cape Cape	**Ghanaian Language**
David Sarpei Nunoo	University of Education, Winneba	
Prof Kwasi Adomako	University of Education, Winneba	
Arko Blay Augustine	University of Education, Winneba	**Government**
Josephine Akosua Gbagbo	Ngleshie Amanfro SHS	
Anitha Oforiwah Adu-Boahen	University of Education, Winneba	**History**
Prince Essiaw	Enchi College of Education	
Blessington Dzah	Ziavi SHTS	**Literature-in-English**
Juliana Akomea	Mangoase SHS	
Dorcas Akosua Opoku	Winneba SHS	**Management in Living**
Grace Annagmeng Mwini	Tumu College of Education	
Eric Abban	Mt. Marys College of Education, Somanya	**Mathematics**
Isaac Buabeng	Mamprobi Sempe 2 JHS	

Benjamin Ofori	Cocoa Research Institute of Ghana Primary School	**Music**
Cosmas W. K. Mereku	University of Education, Winneba	
Davies Obiri Danso	New Juaben SHS	
Prof. Joshua Alfred Amuah	University of Ghana, Legon	
Evans Kofi Mati	GES HQ	
Wilson Klebi	GES HQ	
Professor Emmanuel Obed Acquah	University of Education, Winneba	
Chris Ampomah Mensah	Bolgatanga S.H.S	**Performing Arts**
Doreen Deedee Quarcoo	University of Education, Winneba	
Dr. Latipher Osei Appiah-Agyei	University of Education, Winneba	
Albert Sackey	KNUST SHS	**Physical Education and Health (Core and Elective)**
Ebenezer Ewiah-Quarm	Nkroful Agric SHS	
Evans Asare Yeboah	Adisadel College	
John Benedict Nyarko	Adisadel College	
Mary Aku Ogum	University of Cape Coast	
Dr Linus Kweku Labik	KNUST	**Physics**
Sylvester Affram	Kwabeng Anglican SHTS	
Anthony Mensah	Abetifi Presbyterian College of Education	**RME**
Clement Nsorwineh Atigah	Tamale SHS	
Eliel Keelson	Kwame Nkrumah University of Science and Technology	**Robotics**
Isaac Nzoley	Wesley Girls High School	
Jemima Ayensu	Holy Child Girls SHS, Cape Coast	**Social Studies**
Mohammed Adam, Ph.D.	University of Education, Winneba	
Simon Tengan	Wa Senior High Technical School	
Franklina Kabio-Danlebo	University of Ghana	**Spanish**
Mishael Annoh-Achampong	University of Media, Art and Communication	

Augustina Gloria Henyo	West African Examinations Council	**Inclusive Education**
Dr. Gideon Kwesi Obosu	University of Cape Coast	
Dr. Daniel Fobi	University of Education, Winneba	
Dr. Martha-Pearl Okai	University of Cape Coast	
Francis Kweku Essel	Ashesi University	
Richard Doku	Mampong/Akw SHTS For The Deaf	
Yao Mati Abotsi	Ghana Blind Union	
Seth Kwame Ativor	Okuapemman SHS, Akropong	
Bernard Duorinaah	Presbyterian College of Education	
Evans Odei	Achimota Senior High School	

List of Reviewers			
S/N	**Subject**	**Name of Reviewer**	**Institution**
1	Accounting	Ms. Ivy Eklemet	Univ. of Professional Studies, Accra
2	Additional Mathematics	Francis Tornyenor	Pope John SHS & Minor Seminary, Koforidua
3	Agricultural Science	Selina Acheampong	Univ. College of Agriculture and Environmental
4	Agriculture	Mr. Godwin B. K. Doli	Municipal Educ. Office, Amasaman
5	Arabic	Abdullah Musah Abdullah	Univ. of Media Arts & Communication, UniMAC
6	Art and Design Studio	Ernest Kwasi Elelakpodia	Hope College, Gomoa Fetteh
7	Art and Design Foundation	Ernest Kwasi Elelakpodia	Hope College, Gomoa Fetteh
8	Automotive & Metal Technology	Okyere Kojo Alex	Ada College of Education
9	Aviation & Aerospace Engineering	Dr. Augustine Akuoko Kwarteng	Univ. of Mines and Technology
10	Biology	Rev. Joseph Kweku Delanyo	formerly of Hope College, Gomoa Fetteh
11	Biomedical Science	Dr. Irene Adzo Agbo	Univ. of Professional Studies, Accra
12	Building Construction & Woodwork Technology	Mr. Alex A. Appah	Christ Calvary Prep. Sch.
13	Business Management	Mr. Emmanuel Boakye	Ghana Employers Association

14	Chemistry	Joyce Okreyewa Odei	Hope College, Gomoa Fetteh
15	Clothing and Textiles	Rosemary Quarcoo	Department of Clothing and Textiles, UEW
16	Design & Communication Technology	Esme Serwaa Boateng	Ghana National College, Cape Coast
17	Economics	Dr. Kofi Kamasa	Department of Management Studies, UMaT
18	Electrical & Electronics Technology	Dr. Emmanuel Ampoma Affum	Telecom Engineering Department, KNUST
19	Engineering	Dr. Albert Kotawoke Awopone	Univ. of Skills Training & Entreprenuerial Development
20	English Language	Prof. Philip Gborsong	University of Cape Coast
21	Food and Nutrition	Sylvia Baaba Yankey	
22	French	Gregory Nana Kobina Nketsia Mensah	Mfantsipim School, Cape Coast
23	General Science	Joyce Okreyewa Odei	Hope College, Gomoa Fetteh
24	Geography	Prof Kofi Adu Boahen	University of Education, Winneba
25	Ghanaian Languages	Prof. Regina Caesar	University of Education, Winneba
26	Government	Clement Adu Acheampong	Prempeh College SHS, Kumasi
27	History	Prof Wilson Yayoh	Univertsity of Cape Coast
28	Information Communication Technology	Felicia Archeresuah	Nkoronza Technical Institute
29	Manufacturing Engineering	Dr. Frank Kwabena Nyarko	Kwame Nkrumah Univ. of Science and Technology
30	Mathematics	Ezekiel Osei Bonsu	Bono East Regional Education Office, Nkoranza
31	Management in Living	Ms. Anna Holdbrooke	Swedru Senior High School
32	Music	Prof Mark-Millas Fish	University of Education, Winneba
33	Physical Education and Health Core	Francis Maxwell Ayamba	Atebubu College of Education
34	Physical Education and Health Elective	Sulemana Ahmed Tijani	Atebubu College of Education
35	Performing Arts	Dr. Godfred Asare Yeboah	University of Education, Winneba
36	Physics	Peter Bekoe	Our Lady of Mount Carmel Girls SHS, Techiman
37	Religious and Moral Education	Christain Kwarteng Mensah Oppong	Prempeh College, Kumasi

38	Robotics	Dr. Yaw Okraku-Yirenki	Kofi-Annan Centre of Exellence in ICT
39	Social Studies	Samuel Larbi	St Louis Girls SHS, Kumasi
40	Spanish	Prof. Jonson Asunka	University of Ghana Legon

STRAND: PRINCIPLES OF ROBOTIC SYSTEMS

Sub-Strand: Robots & Society

Weeks 1 & 2

Learning Outcome: *Conceive, evaluate and document high-level robot-based solutions to real-world problems.*

Content Standard: *Demonstrate the ability to relate robots to addressing real societal and industrial problems.*

Week 3

Learning Outcome: *Prepare and justify technical reports for an envisioned solution to a problem and critique technical reports on proposed solutions to known problems.*

Content Standard: *Demonstrate knowledge (DoK) and understanding of the high-level application of robots to address real societal problems.*

INTRODUCTION AND SECTION SUMMARY

Building upon foundational knowledge of robot components and control principles from previous years, this section explores how robotics addresses real-world problems, both locally in Ghana and globally. It guides learners in identifying challenges where robot capabilities can make a positive difference. Furthermore, the section teaches how to meticulously plan the sequence of steps a robot would follow to tackle these problems, utilising technical documentation tools such as flowcharts and pseudocode. This prepares learners to articulate envisioned solutions in a clear, structured manner and to constructively critique them.

The weeks covered by the section are

Week 1

- Identification of both local and global problems that can be fixed using robots and automated systems.

Week 2

- Identification, documentation and presentation of problems in society and local industries and suggesting the role robots can play in solving these problems.

Week 3

1. Preparation of technical reports, including flowcharts or pseudo codes that outline the sequence of steps to solve identified problems.
2. Peer review and critique of prepared technical reports.

SUMMARY OF PEDAGOGICAL EXEMPLARS

A differentiated and inclusive instructional approach guides this section, ensuring varied learner engagement. Lessons begin by building on prior knowledge before specific tasks are undertaken through guided practice. Facilitation covers brainstorming, analysis assistance, instruction on tools like flowcharts and pseudocode, and support for documentation and presentation. Peer interaction for critique is facilitated, and discussions are fostered. Differentiation and inclusivity are integrated throughout all steps to support diverse needs and empower the application and communication of technical concepts.

ASSESSMENT SUMMARY

The following table presents a sample of the key assessments used in this section.

Indicators	DoK Level	Assessment Strategy and/or Mode
Identify both local and global problems that can be fixed using robots and automated systems	4	Short and Extended Response Essays – (Suggested Modes: Individual Classwork, Mid-semester Assessment, End-of-semester Assessment) Group-Based Task/Project – (Suggested Modes: Group project/research/case study)
Identify, document and present on problems in society and local industries and suggest the role robots can play in solving these problems.		Short and Extended Response Essays – (Suggested Modes: Individual Classwork, Mid-semester Assessment, End-of-semester Assessment) Group-Based Task/Project – (Suggested Modes: Group project/research/case study)
1. Prepare technical reports including flowcharts or pseudo codes that outline sequence of steps to solve problems. 2. Peer-review and critique the technical reports.		Short and Extended Response Essays – (Suggested Modes: Individual Classwork, Mid-semester Assessment, End-of-semester Assessment) Individual Practical Task – (Suggested Modes: Individual Project Assessment)

WEEK 1

Learning Indicator: *Identify both local and global problems that can be fixed using robots and automated systems*

FOCAL AREA: FINDING PROBLEMS ROBOTS CAN SOLVE

Introduction

This lesson guides learners to identify real-world problems, both in Ghana and globally. It highlights how the unique capabilities of robots, such as performing dangerous or repetitive tasks, make them suitable for addressing these challenges. The goal is to help learners see where robotic strengths can be applied to create a positive impact.

Building on Foundational Concepts

This section is built upon foundational knowledge from Years 1 and 2. Understanding of robot subsystems and mechanics provides insight into robot capabilities. The concept of feedback systems from Year 2 is used to explain autonomous task performance. This combined understanding allows recognition of specific problem characteristics, like danger or repetition, where robotic capabilities are beneficial. Prior learning on robots' societal impacts highlights the importance of addressing these real-world challenges.

Crucially, problem identification must not occur in an ethical vacuum. Learners must actively use their Year 1 and Year 2 understanding of Roboethics and human-robot coexistence to screen problems. When selecting a challenge, they must evaluate whether a robotic intervention is ethically sound; ensuring it respects human dignity, avoids unfair job displacement without retraining, and maintains safety in public spaces.

Recap Weeks

Year 1	**Week 1** • Analyse how the four organisational performance indicators (price, quality, flexibility, and innovation) have been impacted by the interdependence of humans and robots in working environments. **Week 2** • Identify the economic and social benefits of using robots in 21st-century environments (workplaces, smartly built environments such as smart homes and smart cities, playgrounds, etc.) within the confines of accepted standards and ethics. **Week 3** • Describe robots and identify the differences between robotic and non-robotic systems. • Describe the attributes and functionalities of a robot's subsystems and how they interconnect

Year 2	Week 1 • Analyse and enumerate both the positive and negative impacts of robots on society. • Explain the need for robot coexistence with humans, taking into consideration safety and robotics.

KNOWLEDGE CONTENT

Characteristics of Problems Suited for Robotic Solutions

Robots and automated systems are not typically designed to solve every problem. They are most valuable when applied to tasks or situations that possess specific characteristics, making them difficult, inefficient, or unsafe for human beings. Understanding these characteristics is key to identifying suitable problems for robotic intervention.

Problems well-suited for robotic solutions often involve:

1. **Dangerous Environments:** Robots can operate in places that are hazardous to human health or safety. Examples include working with toxic chemicals, handling radioactive materials, exploring collapsed structures, investigating deep underwater or outer space, or diffusing explosives.
2. **Repetitive Tasks:** Many industrial and logistical processes involve tasks that are highly repetitive and monotonous. Robots excel at performing the same action accurately and consistently for long periods without fatigue or loss of concentration, which can lead to increased efficiency and reduced human error.
3. **High Precision and Accuracy:** Certain tasks require a level of precision or accuracy that is difficult or impossible for humans to maintain consistently. Robots can be programmed to perform movements or manipulations with extremely high precision, critical in fields like manufacturing assembly or microsurgery.
4. **Heavy Lifting and Physical Strain:** Tasks involving lifting or moving heavy objects, or those requiring sustained physical effort that can lead to injury, are often ideal for automation using robots equipped with powerful actuators.
5. **Continuous Operation:** Unlike humans, who require breaks and rest, robots can operate continuously for extended periods, significantly increasing throughput and productivity in industrial or monitoring applications.
6. **Working in Inaccessible Locations:** Robots can be designed to access and work in places that are too small, too remote, or too physically challenging for humans, such as inspecting narrow pipes, exploring caves, or performing maintenance on offshore platforms.
7. **Data Collection and Monitoring:** Robots equipped with various sensors can systematically collect large amounts of data from an environment (e.g., environmental conditions, structural integrity, security surveillance) more consistently and accurately than manual methods.

Applying Robotic Capabilities to Problem Needs

Connecting the characteristics of problems to the fundamental capabilities of robots allows for the identification of specific robotic roles in problem-solving.

1. **Sensing (Year 1 knowledge):** Problems requiring the collection of specific data about the environment (temperature, light levels, presence of objects, chemical composition) can be addressed by robots equipped with the relevant sensors. For instance, monitoring pollution levels in a water body or detecting gas leaks in a mine.
2. **Movement and Navigation (Year 1 components, Year 2 control):** Problems requiring movement through complex or dangerous terrain, covering large areas, or accessing confined spaces can be solved by robots designed for mobility (wheeled, legged, aerial, underwater) and programmed for navigation. Examples include delivering supplies in disaster zones or inspecting pipelines.
3. **Manipulation and Interaction (Year 1 components, Year 2 control):** Problems requiring physical interaction with objects – lifting, sorting, assembling, cutting, welding – can be handled by robots equipped with manipulators (robot arms, grippers) controlled precisely. This is common in manufacturing and waste processing.
4. **Computation and Decision Making (Year 2 programming/control):** Problems requiring quick analysis of data and autonomous decision-making based on that data leverage the robot's processing power and programmed logic (flowcharts, pseudocode, feedback loops from Year 2). An autonomous vehicle reacting to traffic or a monitoring robot deciding to alert authorities based on sensor readings are examples.

By considering the task a robot needs to perform within a problematic situation and linking that task back to fundamental robotic capabilities and the principles of control, one can effectively identify where robotic intervention is feasible and beneficial.

Examples of Robotic Applications in Problem-Solving

Robots are already being used globally and increasingly in Africa to address a wide array of problems:

1. **Agriculture:** Robots for planting, harvesting, monitoring crop health (using sensors and cameras), precision irrigation, and weeding can improve efficiency, reduce labour costs, and increase yields, particularly relevant for local farmers.
2. **Healthcare:** Robotic surgery systems allow for minimally invasive procedures with greater precision. Robots can also assist with patient care, rehabilitation, or delivering medication within hospitals.
3. **Industry and Manufacturing:** Robots perform assembly, welding, painting, and quality inspection tasks, improving speed, consistency, and safety in factories and workshops.
4. **Environment:** Robots are used for monitoring pollution in air, water, and soil, collecting waste (e.g., plastic from oceans), performing environmental inspections in hard-to-reach areas, or assisting in conservation efforts.

5. **Safety and Security:** Robots are deployed for surveillance, inspecting dangerous structures (like bridges or buildings after disasters), assisting in search and rescue operations, or handling hazardous materials. Examples include using drones for monitoring illegal mining sites or for security surveillance.
6. **Logistics and Transportation:** Automated guided vehicles (AGVs) move goods in warehouses, and autonomous drones can deliver packages to remote locations.

Identifying problems suitable for robots involves looking at challenging tasks in various sectors and determining if robotic capabilities offer a viable and advantageous alternative to current methods. This requires an understanding of both the problem's demands and the robot's potential.

Learning Tasks

The following tasks are designed to help learners actively identify problems suitable for robotic intervention. Depending on the available time or resources, administer one or more of the following learning tasks to help learners reinforce understanding and acquire new knowledge or skills.

1. In small groups, generate a list of problems observed in your immediate community, school, or anywhere within Ghana. Consider problems related to areas such as waste management, agriculture, healthcare access, safety, traffic, or the environment. As you list problems, think about the characteristics of problems suited for robotic solutions, such as those where the required task might be dangerous for humans, necessitate working in difficult or inaccessible environments, or involve actions that must be done repeatedly or with precision.
2. Working with your group, generate a list of problems occurring in other parts of the world. Consider issues such as disaster response, the exploration of hazardous environments, tasks within large-scale manufacturing processes, or the challenge of providing aid in remote locations. Think about global problems that share characteristics with the local problems discussed in Task 1, where robotic capabilities might be relevant. Utilise resources such as news articles or videos about global issues to help develop ideas.
3. Review the combined list of local and global problems generated in Tasks 1 and 2 and carry out the following tasks:
   a. Group similar problems together into relevant categories (for example, environmental issues, health challenges, industrial tasks, safety concerns).
   b. For each problem listed, discuss and determine why a robot or automated system might be a suitable tool for assistance. State the *specific kind of task* or *action* a robot could perform to help address the problem. (It is not necessary to design the robot itself at this stage.)
   c. Provide a brief explanation next to each problem, stating why the capabilities of robots make them potentially relevant in that specific situation.

Hint

- *When thinking about robotic relevance, consider tasks that are repetitive, dangerous, require strength, precision, or operating in places where humans cannot easily go.*

PEDAGOGICAL EXEMPLARS

Recognise that learners possess shared characteristics while also having individual differences in interests, readiness, and learning styles. Plan to address these variations. Provide multiple avenues for learners to engage with the lesson content, adjusting the level of detail or support as needed to help all learners achieve the lesson outcomes. Employ strategies such as Think-Pair-Share to encourage participation.

1. **Step 1: Introducing the Lesson and Activating Prior Knowledge**

   Begin by referencing the learners' knowledge of basic robot components and functions from Year 1 and Year 2. Discuss how these capabilities enable robots to perform specific tasks. Introduce the lesson's primary focus: identifying problems that robots could help solve. Ensure the language used is clear and accessible to all learners; rephrase complex terms if necessary.

To ensure differentiation and inclusivity

a. Utilise visual aids, such as diagrams of robots and their components, to support understanding.

b. Ask simple recall questions to activate knowledge from previous years (e.g., "What is the function of a sensor?").

c. Provide sentence frames or prompts to assist learners in verbal contributions (e.g., "A robot might help with [problem] because it can perform [robotic action]").

d. Ask more complex questions requiring learners to explain how different components might work together in a simple system (e.g., how sensors and actuators function collaboratively for navigation).

e. Verify that visual aids are clear and easily seen. Provide key terms in written format for learners with hearing impairments. Offer various methods for learners to participate, including verbal responses and written notes.

2. **Step 2: Facilitating Problem Brainstorming**

   Guide groups as they undertake the task of brainstorming local problems (Learning task 1) and global problems (Learning task 2). Foster an environment where learners feel comfortable sharing any idea. Circulate among the groups to monitor their progress and offer prompts to stimulate thinking, particularly encouraging consideration of local Ghanaian contexts. Share diverse examples of problems to broaden their perspective on different types of challenges.

To ensure differentiation and inclusivity

a. Offer predefined categories or themes to assist groups in structuring their brainstorming process (e.g., problems at home, school, or in the environment).

b. Challenge some groups to consider less apparent problems or issues that involve multiple contributing factors.

c. Arrange group compositions to allow for different roles (such as a recorder, a speaker, or an idea generator), accommodating various learner strengths.

d. Maintain clear and simple language when providing instructions for the tasks.

3. **Step 3: Guiding Categorisation and Relevance Discussion**

Direct groups through the task of reviewing and categorising their brainstormed problems (Learning Task 3). Assist them in grouping similar issues logically. For each identified problem, prompt discussion by asking questions such as "Why might a robot be helpful here?" or "What specific task could a robot perform to address this?" Guide them in connecting their ideas about robotic roles back to the capabilities of robots they have learned about in previous years.

To ensure differentiation

a. Provide sample categories to help initiate the categorisation process.

b. Work closely with groups that find it challenging to articulate the reason for robotic intervention, helping them link the problem's needs to specific robotic capabilities (e.g., "This problem involves lifting heavy items, and actuators provide strength, making a robot suitable").

c. Suggest that more advanced groups consider whether a complex problem might require multiple types of robots or a system of automated units.

4. **Step 4: Facilitating Class Sharing and Summary**

Invite groups to share some of the problems they identified and the potential robotic roles they discussed. Facilitate a brief class discussion following the presentations. Summarise the wide range of problems that robots and automated systems can potentially help address, drawing examples from both local and global contexts.

To ensure inclusivity

a. Actively encourage different groups to share their ideas, ensuring that contributions are not limited to the most vocal learners.

b. Utilise visual summaries, perhaps written on a board or displayed digitally, to reinforce the key problems and potential robotic roles discussed.

KEY ASSESSMENT LEVEL 4: INDIVIDUAL CLASS EXERCISE

Short and Extended Response Essays (10 marks)

1. **Instructions:** Answer the following questions with clear explanations and justifications for your reasoning. **(DoK 3)**
   - **a.** Describe one specific problem in a local Ghanaian industry where a robot could potentially improve efficiency or safety. Explain why this problem is well-suited for a robotic solution (referencing at least two physical characteristics), and justify how your proposed solution addresses at least one key ethical concern (such as safety, transparency, or user acceptance). (4 marks)
   - **b.** Consider the global problem of cleaning up plastic waste from the surface of the ocean. Identify one specific task a robot could perform as part of a solution. Justify why a robot is more suitable for this specific task than a human, focusing on the challenges of the ocean environment. (3 marks)
   - **c.** Explain how the robotic capability of 'Sensing' combined with the concept of 'Continuous Operation' makes robots particularly useful for monitoring environmental conditions over long periods in remote locations. Provide a specific example. (3 marks)
2. **Instructions**: Read the scenario and answer the questions. The questions are designed to encourage extended critical thinking and reasoning. **(DoK 4)**

Scenario: *In many rural or remote regions in Ghana, maintaining the structural integrity of tall radio masts presents significant challenges. Traditionally, human technicians conduct inspections, often involving dangerous and logistically complex procedures.*

   - **a.** Explain the specific risks and logistical issues associated with human-led inspections of tall radio masts.
   - **b.** Evaluate how robotic systems could address these challenges. In your evaluation, compare at least two robotic capabilities (e.g. mobility, sensing, data processing) with current human methods.
   - **c.** Critically assess any limitations or trade-offs that might arise from replacing human inspection with robotic intervention.
   - **d.** Conclude by justifying whether you believe the robotic approach represents a sustainable and ethical improvement for this local challenge.

3. ***Group-Based Task/Project DoK 4 (20 marks)***

Project Title: *Robotic Solutions for Local Challenges*

Task Description

Your group is to identify a significant problem in your local community or school that could be addressed using a robotic or automated system. Prepare a report or presentation that includes the following:

a. Problem Identification and Impact (4 marks)

i. Describe the problem clearly and in detail.

ii. Explain its impact on individuals, the school, or the wider community.

b. Problem Analysis for Robotic Suitability (4 marks)

i. Analyse the problem by identifying:

- Repetitive or predictable tasks
- Need for precision, speed, or safety
- Environmental conditions

ii. Justify why the problem is suitable for automation or robotics.

c. Proposed Robotic Solution (4 marks)

i. Clearly state the role or function of the robot.

ii. Explain how the robot will help solve or reduce the problem while adhering to roboethics..

d. Conceptual Design of the Robot (6 marks)

(This addresses the reviewer's concern directly) Your group must present a **basic design** of the proposed robotic system, including:

i. A **labelled diagram or sketch** of the robot/system

ii. Identification of key components, such as:

- Sensors (e.g., ultrasonic, touch, light)
- Actuators (e.g., motors, servos)
- Control unit (e.g., microcontroller like Arduino)

iii. A simple **input–process–output (IPO)** or system flow diagram

iv. Description of how the system operates (step-by-step logic or pseudocode)

Note: This is a **conceptual design**, not a physical build.

e. Feasibility and Reflection (2 marks)

i. Discuss:

- Practicality of implementing the design in your local context
- Possible challenges (cost, power, maintenance, skills)
- Suggested improvements or future enhancements

Conclusion

This week's lesson built upon previous knowledge of robot fundamentals and control systems. Focus was placed on recognising real-world problems, both within Ghana and internationally, where robot capabilities can be applied. Problems possessing characteristics such as occurring in dangerous environments or requiring high precision were identified as particularly suited for robotic assistance. Understanding how robot strengths align with these challenges is key to seeing how automated systems can contribute positively to society.

PLC Session 1

45 minutes	Planning & Reviewing the Learning Plan

1. Begin the session by recapping lessons from the previous year/session/week, focusing on how learning was applied.
2. Use your subject-specific app to explore the content before planning. Ask the app to
   - explain the week's concepts in a practical way that will make it easier for learners to understand
   - identify two or three common misconceptions learners may have on the week's content and how to address them using contexts familiar to Ghanaian learners. (NTS 3a, 3g and 3m).
3. Use your subject-specific app to plan your lesson. Ask the app to
   - suggest the most effective method for teaching the week's concepts to learners with diverse backgrounds.
   - suggest learning activities, differentiation and essential questions based on the method of teaching (NTS 3a-3q)
   - review your prepared learning plan and address any challenging issues you may have identified in your subject area (NTS 3a-3c).

Note

- Take note of the recap topics linked to Years 1 and 2 while preparing for your lesson.
- Refer to the Values Learning Community Handbook and link the values for the week in your lesson to ensure that learners are applying what they learn.
- Plan the week's key assessment with your app

- The key assessment for the week is ***individual class exercise***. Chat with your app to plan this assessment, indicating what you and your learners will do before, during and after the exercise (NTS 3k-3q).

10 minutes	Content Exploration

4. As a subject group, select one problem or idea/term in the week's content that is challenging (not a routine procedural question, but one that requires conceptual understanding) and then

- use the app, ask it to solve the problem step by step and explain the reasoning behind each step (you may be assisted by a teacher who is strong in the content area)
- all teachers in the group work through a similar problem individually for 3–4 minutes
- the group then works through the problem together step by step
- highlight the most common errors made at each step and discuss (NTS 1c, 1e, 2c, 2e, 3e-3j).

Note

- This part is not just about 'How do I teach this?' but how do I teach it for learners to understand?
- If the group still shares the same content gap, use the app: ask it to solve the problem step by step and explain the reasoning behind each step.
- Problems/ideas discussed should be sourced from the Teacher Manual content, not invented separately. This keeps the content exploration directly connected to what teachers will deliver.
- Enact a teaching activity based on your learning plan (NTS la-1b, 1e-lg, 2c, 2e, 2f, 3c-3j and 3o).

30 minutes	Enactment

5. Colleagues observe with specific attention to clarity of concept explanation, use of familiar Ghanaian examples and how misconceptions are handled if they arise.
6. After the enactment, ask for structured feedback from the group
   - was the concept explained in a way that a learner with no prior knowledge could follow?
   - were the examples grounded in contexts familiar to Ghanaian learners?
   - did any misconceptions surface and if so, how were they addressed?

5 minutes	Reflection

7. Reflect and share your views on the session. One thing that worked well, what did not work well and what you will do differently. (NTS 1a, 1b).
8. Identify a critical friend to observe and provide feedback on your lesson in relation to the PLC Session conducted (NTS 1f, 1g).
9. Remember to:
   - transfer your chats into your learning planner template.
   - read your Teacher Manual and related Learner Material in preparation for the next session. Take note of the recap weeks, which are linked to Years 1 and 2 (NTS 3a).
   - integrate National Values, GESI, SEL, and 21st century skills into your learning activities, assessment tasks, and rubrics/mark schemes.

WEEK 2

Learning Indicator: *Identify, document and present on problems in society and local industries and suggest the role robots can play in solving these problems*

FOCAL AREA: REPORTING ROBOT IDEAS FOR LOCAL CHALLENGES

Introduction

Building on prior problem identification skills, this lesson focuses on understanding specific challenges in Ghanaian society and local industries. Rigorous analysis of these problems is undertaken, followed by determining the precise role a robot could perform. Skills in properly documenting and presenting these findings and suggested robotic roles are developed, serving as an essential first step towards effectively pitching ideas for solving problems identified within their own communities.

Building on Foundational Concepts

This lesson builds upon the foundation laid in Years 1 and 2. Understanding robot subsystems and basic mechanics from Year 1 allows learners to define a robot's potential role based on its physical abilities. Knowledge of planning logic using IPOs, algorithms, flowcharts, and pseudocode from both years is applied when outlining the robot's contribution. Prior experience with structured troubleshooting reports provides the basis for documenting problem analysis and the proposed robotic role. Furthermore, insights into robots' societal impacts from earlier studies inform the analysis of problems and the justification of solutions.

Recap

Year 1	**Week 2** • Identify the economic and social benefits of using robots in 21st-century environments... within the confines of accepted standards and ethics. **Week 3** • Describe the attributes and functionalities of a robot's subsystems and how they interconnect **Week 17** • Determine the Inputs, processes, and outputs required to solve a particular problem. • Define solutions to basic automated and robotic problems using algorithms, pseudocodes, and flowchart diagrams. **Weeks 19&20** • Create robots using robotic kits and/or local materials to implement basic mechanics for actuation..." (specifically "ARMS, WINGS & OTHERS" – Gripping Figures, Lifting Mechanisms)

Year 2	**Week 1** • Analyse and enumerate both the positive and negative impacts of robots on society. • Explain the need for robot coexistence with humans, taking into consideration safety and robotics. **Week 3** • Utilise component and system diagrams to design a feedback system for implementing controls in given scenarios. **Week 17** • Define solutions for control and feedback systems using algorithms, pseudocodes, and flowchart diagrams. **Weeks 19 & 20** • Learners should create robots using robotic kits and/or local materials to implement basic mechanics for actuations that make use of…" (specifically "ARMS, WINGS & OTHERS" – Shooting robots, Automatic Doors, Raking up or out, Creating Wind) **Week 23 & 24** • Observe and identify flaws in an operational robot and trace them to either algorithm flaws, design flaws or coding errors. • Use a series of iterations to fix the identified flaws or improve the performance of solutions.

KNOWLEDGE CONTENT

Understanding the Problem in Depth

Moving beyond simply identifying a problem involves a thorough analysis to understand its complexities. A deep analysis of a problem in Ghanaian society or a local industry requires examining several facets:

1. **Problem Definition:** Clearly stating the problem in specific terms. What exactly is the issue? Where does it occur? Who is affected?
2. **Root Causes:** Investigating the underlying reasons for the problem. What factors contribute to its existence and persistence? Are there multiple causes?
3. **Impact Analysis:** Assessing the effects of the problem. How does it impact individuals, the community, the environment, or the industry? What are the consequences of inaction?
4. **Existing Solutions:** Identifying any current methods or attempts to address the problem. Why are these existing solutions insufficient or ineffective? Understanding limitations helps justify the need for a new approach.

Conducting this analysis provides a comprehensive picture of the challenge, which is necessary before considering a potential robotic intervention. Resources such as community observations, interviews with affected individuals or industry workers, local news reports, and online data can provide valuable information for this analysis.

Defining the Robot's Specific Contribution (The Robotic Role)

Once a problem is analysed in depth, the next step is to define the specific task or function a robot or automated system could perform to help address it. This is about outlining the *role* of the robot within the context of the problem, not designing the complete technical system.

Defining the robot's role involves:

1. **Identifying Key Tasks:** Pinpointing the specific actions within the larger problem that a robot is capable of performing effectively (drawing on knowledge of robot capabilities like sensing, movement, manipulation, computation from Year 1 and Year 2).
2. **Matching Capabilities to Needs:** Linking the requirements of the identified task directly to the strengths of robotic systems (e.g., the task requires operating in a dangerous area, aligning with a robot's ability to work in hazardous environments; the task involves lifting heavy items, aligning with a robot's strength from actuators). This explains why a robot is a suitable tool for this particular task.
3. **Stating the Role Clearly:** Articulating the robot's function as a clear statement or brief description. For example, if the problem is waste accumulation in gutters, the robot's role might be "autonomous collection of solid waste from drainage systems." If the problem is inefficient sorting of produce, the role could be "sorting ripe and unripe fruits based on colour and size."

Defining the role requires applying knowledge of robotic capabilities and understanding the characteristics of problems suitable for robots to the specific details of the analysed problem.

The Purpose of Structured Reporting for Problems and Solutions

Effective communication of ideas is vital in technical fields. Just as a structured report was used in previous years to document the process of troubleshooting and fixing a robot (Recalling the components of a technical report from Year 2, such as the Title Page, Table of Contents, and a logical flow from problem description to solution and conclusion), a similar formal approach is necessary to clearly articulate a real-world problem and a proposed robotic role as a potential solution.

A well-structured report serves as a clear record of analysis and proposals, making it easier for others to understand the identified problem and the potential contribution of a robot. This documentation is a critical first step in communicating technical ideas, particularly when seeking support or collaboration. The format adapts the principles of formal reporting to the specific task of problem analysis and robotic role definition.

The Purpose of Structured Reporting for Problems and Solutions

Effective communication of ideas is vital in technical fields. Just as a structured report was used in previous years to document the process of troubleshooting and fixing a robot (Recalling the components of a technical report from Year 2, such as the Title Page, Table of Contents, and a logical flow from problem description to solution and conclusion), a similar formal approach is necessary to clearly articulate a real-world problem and a proposed robotic role as a potential solution.

A well-structured report serves as a clear record of analysis and proposals, making it easier for others to understand the identified problem and the potential contribution of a robot. This documentation is a critical first step in communicating technical ideas, particularly when seeking support or collaboration. The format adapts the principles of formal reporting to the specific task of problem analysis and robotic role definition.

The Adapted Report Structure: Documenting Problems and Robotic Roles

This adapted structure provides a framework for documenting the findings from problem analysis and the proposed role a robot could play.

1. **Title Page**
   a. **Content Focus:** This page introduces the report. It should clearly state the main subject, which is the specific real-world problem analysed. It should also identify the authors of the report.
   b. **Knowledge Point:** Emphasise the importance of a clear title that captures the essence of the problem being addressed. Standard elements like the date and authors' names lend professionalism.
2. **Table of Contents**
   a. **Content Focus:** A list showing the main sections and any sub-sections of the report, along with their corresponding page numbers.
   b. **Knowledge Point:** Explain that this provides a roadmap for the reader, allowing quick access to different parts of the analysis and proposal.
3. **Introduction**
   a. **Content Focus:** This section sets the stage. It should provide a concise overview of the specific societal or local industry problem that was analysed. It introduces the general area of concern and briefly states the report's purpose, which is to detail the problem and propose a potential role for a robot or automated system in addressing it.
   b. **Knowledge Point:** The introduction should clearly define the scope of the problem being discussed in the report. It should capture the reader's attention regarding the significance of the problem.
4. **Problem Context**
   a. **Content Focus:** This section provides background information necessary to understand the problem fully. It should describe the environment or setting where the problem occurs (e.g., a specific farming area, a local market, a particular type of workshop). It should also briefly outline the current process or situation related to the problem before any potential robotic intervention.
   b. **Knowledge Point:** Understanding the context is vital for evaluating the feasibility and relevance of any proposed solution. Describing the current process helps highlight where inefficiencies or difficulties lie.

5. **Problem Analysis & Suitability Justification**

   a. **Content Focus:** This is a core section detailing the findings from the deep problem analysis. It should describe the root causes of the problem and its significant impacts on the community or industry. Crucially, this section must justify why a robot or automated system is a suitable tool for addressing this particular problem or specific tasks within it. This justification involves linking the characteristics of the problem (e.g., it occurs in a dangerous location, involves repetitive tasks, requires high precision) to the relevant capabilities of robots (e.g., ability to operate in hazardous environments, perform tasks continuously, execute precise movements - Referencing Year 1 knowledge of components and Year 2 knowledge of how components enable capabilities).

   b. **Knowledge Point:** This section moves beyond simple description to analysis and reasoned argument. Learners must demonstrate their understanding of both the problem's nature and the robot's potential strengths by explicitly connecting them.

6. **Proposed Robotic Role**

   a. **Content Focus:** This section clearly defines the specific task or function that a robot or automated system would perform to help mitigate or solve the analysed problem. This should be a clear statement outlining *what* the robot would do, not *how* it would be built in detail.

   b. **Knowledge Point:** The role should be specific and directly related to the problem's needs identified in the analysis. It represents the envisioned contribution of automation.

7. **Potential Impact and Benefits**

   a. **Content Focus:** This section describes the anticipated positive changes or improvements that could result from the robot performing its proposed role. It should explain how the robot's actions would help address the problem's impacts or improve the current situation (e.g., increased safety, improved efficiency, reduced waste, better monitoring).

   b. **Knowledge Point:** Articulating the potential benefits helps demonstrate the value of the proposed robotic intervention and reinforces the justification for its use.

8. **Potential Challenges, Risks, and Mitigations**

   a. **Content Focus:** This section outlines the realistic operational challenges and potential risks associated with implementing the proposed robotic solution. It covers physical/technical risks (e.g., dust damage to camera lenses, erratic power supply), economic risks (e.g., high initial cost, local technical maintenance skills gap), and social/ethical risks (e.g., workforce anxiety, safety of bystanders). Crucially, for every identified risk, students must propose a realistic engineering or operational mitigation strategy (e.g., designing protective enclosures, adding solar/battery backups, or proposing human retraining programs).

   b. **Knowledge Point:** Engineering design requires balancing optimism with extreme realism. Actively designing mitigations for potential failure points is the direct application of Roboethics and is a standard industry practice in technical reporting.

9. **Conclusion**
   a. **Content Focus:** A brief concluding statement that summarises the key points of the report. It should restate the problem addressed, the specific role proposed for the robot, and the main potential benefit of this intervention.
   b. **Knowledge Point:** The conclusion provides a concise summary of the findings and proposal, leaving the reader with a clear understanding of the core idea.
10. **Appendices**
   a. **Content Focus:** This section can include any supplementary materials that support the report but are not essential to the main text. This might include diagrams of the problem scenario, simple sketches of the robot's form related to its task (e.g., showing a gripper for collection), photos related to the problem, or data collected during analysis.
   b. **Knowledge Point:** Appendices provide supporting evidence and visual aids that enhance the report's clarity and credibility.

Presenting Technical Ideas Effectively

Communicating the findings and proposed robotic role verbally is a skill necessary for sharing ideas with others, analogous to pitching a concept. Effective presentation involves:

1. **Structuring the Presentation:** Organising the information logically, often following the structure of the technical report to ensure a coherent flow from problem to proposed solution.
2. **Clarity and Conciseness:** Presenting information using clear, simple language, avoiding overly technical jargon or explaining it clearly if necessary. The goal is to be understood by the intended audience.
3. **Using Visual Aids:** Employing slides, diagrams, images, or short videos to support the verbal presentation, making it more engaging and helping to explain complex aspects of the problem or the proposed role.
4. **Engaging the Audience:** Speaking clearly and at an appropriate pace, maintaining eye contact, and showing enthusiasm for the subject matter. Being prepared to answer questions demonstrates confidence and understanding.
5. **Highlighting Impact:** Clearly articulating why the chosen problem is significant and why the proposed robotic role is a suitable and beneficial approach, reinforcing the justification from the report.

Recall from Year 2

Remember when we discussed the potential impact of robots on society and ethical considerations? (Refer to Year 2, Section 1). As we think about where robots can play a role in solving problems, it is important to also consider the people involved and the wider consequences.

Learning Tasks

These tasks require learners to analyse a specific local problem, define a potential role for a robot, document their findings and proposal in a structured report, and present their ideas. Depending on the available time or resources, administer one or more of the following learning tasks to help learners reinforce understanding and acquire new knowledge or skills.

1. Working individually or in small groups, select one problem that is relevant to Ghanaian society or a local industry where you believe a robot or automated system could potentially help solve the problem.
   - Conduct a thorough analysis of the selected problem. This involves investigating its root causes, understanding its effects on individuals, the community, or the industry, identifying who is affected, and examining why any existing solutions are insufficient or ineffective.
   - Document the key findings from your problem analysis.
2. Based on your analysis of the problem, define the specific role or function that a robot or automated system could perform to address it.
   - Clearly state what specific task(s) the robot would need to perform.
   - Explain why this specific robotic action is valuable in the context of the problem. Connect the required tasks to the capabilities of robots you understand from your studies (Year 1 and Year 2 knowledge).
3. Prepare a structured technical document/presentation outlining your work on the selected problem and the proposed robotic role. The document/presentation should follow a clear structure, similar to the adapted report format used for documenting problems and robotic roles. Ensure your document/presentation includes:
   - A clear description of the chosen problem and its impact.
   - A summary of the key findings from your problem analysis.
   - A clear statement defining the suggested role(s) the robot would play.
   - A justification explaining why a robot is suitable for this role, referencing the problem's characteristics and relevant robotic capabilities.
   - Detail at least two realistic challenges or risks (technical, economic, or social) your robot might face when deployed in Ghana, and explain how your design or implementation plan actively mitigates these risks.
   - (Optional) Include visual aids, such as simple diagrams, to support your explanations.

   Prepare the document/presentation using clear, concise language and ensure it is well-organised following the chosen structure.
4. Prepare and deliver a short presentation to the class summarising your chosen problem, the key findings from your analysis, and the suggested role for a robot or automated system.

- Create visual aids, such as slides or a poster, to support your presentation.
- Practice your presentation to ensure it is delivered clearly and concisely.

Things to Note

Documentation and presentation are key skills. Practice makes perfect! Ensure your document is well-organised and your presentation is clear and easy to follow. Consider using diagrams to help explain complex ideas.

PEDAGOGICAL EXEMPLARS

Consider the shared characteristics and individual differences among learners, including their interests, readiness, and learning styles. Plan instructional approaches that address these variations. Offer multiple ways for learners to engage with the content and demonstrate their understanding, ensuring all learners can achieve the stated learning outcomes. Strategies like Think-Pair-Share can be used to promote active participation.

1. **Step 1: Recap and Transition to New Focus**

   Briefly review the key concepts from the previous lesson, particularly the skills learned in identifying problems where robots could be helpful. Introduce the activities for this lesson, highlighting the shift towards analysing a specific local problem in depth, defining a precise robotic role, and documenting and presenting these findings. Emphasise that the focus is on articulating what the robot would do and why it is suitable, rather than designing its technical details.

To ensure differentiation and inclusivity

- **a.** Use visual prompts related to the previous lesson's problem identification activity.
- **b.** Clearly state the learning objectives for the current lesson, providing them verbally and in written format.

2. **Step 2: Facilitating Problem Selection and Analysis**

   Guide learners in selecting a specific local problem that is manageable for in-depth analysis (Learning task 1). Provide relevant resources to support their research and analysis. Circulate among the groups to help learners break down the problem into its key aspects and identify contributing factors and impacts. Encourage them to consider the human element and any factors unique to the local Ghanaian context.

To ensure differentiation

- **a.** Offer a pre-selected list of potential local problems with some initial background information for learners who need a starting point.
- **b.** Provide a structured worksheet with guiding questions to help learners conduct their analysis systematically.

c. Challenge more advanced learners to select problems that require analysing interconnected issues or considering multiple perspectives.

3. **Step 3: Guiding the Definition of the Robot's Role**

   Direct learners through the process of defining the specific role a robot could play based on their problem analysis (Learning task 2). Help them translate the identified needs of the problem into specific actions or functions a robot could perform. Prompt them to connect these required actions back to their knowledge of robot capabilities (such as sensing, movement, or manipulation), drawing upon concepts from Year 1 and Year 2. Guide them in articulating why a robot is suitable for this specific role when compared to non-robotic approaches.

To ensure differentiation

a. Provide clear examples illustrating how problem needs can be translated into specific robotic actions and justified by capability (e.g., linking the need for heavy lifting to actuator strength).

b. For learners who need additional support, work with them to draft initial statements of the robot's role and justification.

c. Challenge learners to consider if the proposed role requires coordination between multiple robots or interaction with people, and how their Year 2 knowledge of control systems might influence the design considerations for that role.

4. **Step 4: Supporting Documentation**

   Guide learners as they prepare their structured technical document outlining their problem analysis and proposed robotic role (Learning task 3). Provide a clear template or outline for the document structure, aligning with the adapted report format discussed in the knowledge content. Offer support with the writing process, focusing on articulating technical information clearly and concisely. Emphasise the importance of clear language and logical organisation.

To ensure inclusivity

- Provide sentence starters or standard phrases for different sections of the document.
- Allow learners to work collaboratively in pairs or small groups on the documentation.
- Offer opportunities for learners to receive feedback on drafts of their document from peers or the teacher.
- Challenge learners to incorporate simple diagrams, sketches, or flowcharts into their document to visually represent aspects of the problem or the robot's role.

5. **Step 5: Guiding Presentation Preparation**

   Direct learners in preparing for their short presentations (Learning task 4). Discuss effective strategies for presenting technical information clearly and engagingly, including techniques for using visual aids and managing presentation time. Allocate class time for learners to prepare their visual aids and practise delivering their presentations.

To ensure inclusivity

- **a.** Provide a simple template for presentation slides or posters.
- **b.** Allow learners to present in pairs or small groups, enabling them to share presentation responsibilities.
- **c.** Challenge more advanced learners to consider how to engage their specific audience or prepare for potential questions.
- **d.** For learners who may have communication challenges, allow alternative presentation formats such as pre-recorded videos or detailed poster presentations accompanied by a brief summary. Ensure presentation areas are accessible for all learners.

6. **Step 6: Facilitating Presentations and Discussion**

   Manage the presentations by each group or individual. Facilitate a class discussion after each presentation, encouraging learners to ask clarifying questions and provide constructive feedback to their peers. Guide the discussion to highlight well-analysed problems and clearly articulated robotic roles.

To ensure inclusivity

- **a.** Establish clear guidelines for respectful listening and questioning during presentations.
- **b.** Ensure that question-and-answer sessions are managed in a way that encourages participation from all learners.

KEY ASSESSMENT: GROUP PROJECT

Hint

- *Assign Group Project to learners and give them reasonable time to complete it. Use the SS App to generate the project as well as the rubric. Learners' projects should be marked immediately and scores entered into the SAP.*

1. **Project Title:** Designing a Robotic Solution Concept for a Local Industry Efficiency Challenge

Task

Your group is to identify a specific, complex efficiency challenge within a local Ghanaian industry (e.g., small-scale processing, logistics, agriculture, craft production). Conduct a thorough analysis of this challenge and design a high-level conceptual robotic system to address it. Prepare a detailed technical report and a presentation outlining your work. Your report and presentation should include the following, structured according to the adapted report format:

- **a.** **Introduction:** Overview of the industry and the specific efficiency problem addressed.
- **b.** **Problem Context:** Description of the environment and the current inefficient process.
- **c.** **Problem Analysis & Suitability Justification:** In-depth analysis of the problem's root causes, impact, and existing solutions. Justification why a robotic solution is suitable,

linking problem characteristics (e.g., repetitive, requires precision, dangerous) to relevant robotic capabilities (sensing, movement, manipulation) and principles (e.g., continuous operation, accuracy).

d. **Proposed Robotic Role & Conceptual System:** Define the specific task(s) the robot would perform. Describe, at a high level, the conceptual robotic system needed (e.g., type of robot, key sensors, actuators, basic operational idea) to perform this role, showing how components enable the task.

e. **Potential Impact and Benefits:** Explain how the proposed robotic role improves efficiency and addresses the problem's negative impacts compared to the current method.

f. **Evaluation of Challenges:** Discuss potential challenges in implementing this conceptual prototype in the real local industry context (e.g., cost, power, maintenance, required skills, social acceptance).

g. **Conclusion:** Summarise the problem, proposed role, and potential benefits.

h. **Appendices:** Include diagrams of the problem scenario, sketches of the conceptual robot, or relevant data/research findings.

(***30 marks***)

2. **Individual Practical Task - (Suggested Modes: Individual Project Assessment)**

Project Title: Conceptual Robotic System Design for Community Problem

Task

Identify a significant problem in your immediate school environment or local community (not an industry) that you believe could be partially addressed by a robot. Design a high-level *conceptual* prototype of a feedback-driven robotic system to help mitigate this problem. Document your analysis, design, and justification in a detailed report (following the adapted structure) or a comprehensive presentation. Your report or presentation must include:

a. Detailed analysis of the specific community problem, explaining its causes, impact, and why it needs addressing.

b. Justification for *why* a robot is a suitable tool for this problem, linking its characteristics to robotic capabilities.

c. Description of your conceptual robotic system, defining its task(s) and identifying the *type* of sensors and actuators it would need (linking to Year 1 knowledge).

d. Explanation of *how* the system would use *feedback* from its sensors to perform its task autonomously (linking to Year 2 feedback knowledge).

e. Discussion of the potential impact and benefits of this system on the community.

f. Evaluation of potential challenges in implementing this conceptual system in the real community environment (e.g., cost, adherence to roboethics, power, user acceptance, environmental factors).

(***25 marks***)

Conclusion

Moving beyond problem identification, focus in this lesson was placed on the in-depth analysis of specific challenges within local communities and industries. The task of defining a precise role a robot could perform to assist with these problems was undertaken. Skills in properly documenting findings and suggested robotic roles were developed. These abilities are considered essential for clearly communicating technical ideas and serving as a crucial step towards pitching solutions for problems encountered in learners' own communities.

PLC Session 2

45 minutes	Planning & Reviewing the Learning Plan

1. Begin the session by recapping lessons from the previous session, focusing on how learning was applied.
2. Use your subject-specific app to explore the content before planning. Ask the app to
   - explain the week's concepts in a practical way that will make it easier for learners to understand.
   - identify three common misconceptions learners may have on the week's content and how to address them using contexts familiar to Ghanaian learners (NTS 3a, 3g and 3m).
3. Use your subject-specific app to plan your lesson. Ask the app to
   - suggest the most effective method for teaching the week's concepts to learners with diverse backgrounds.
   - suggest learning activities, differentiation and essential questions based on the method of teaching (NTS 3a-3q).
   - review your prepared learning plan and address any challenging issues you may have identified in your subject area (NTS 3a-3c).

Note

1. Take note of the recap topics linked to Years 1 and 2 while preparing for your lesson.
2. Refer to the Values Learning Community Handbook and link the values for the week in your lesson to ensure that learners are applying what they learn.
3. Use your app to plan and administer the mode of assessment for the Student Assessment Portal (SAP).
   a. The key assessment for the week is a ***group project***. Chat with your app to plan this assessment by reviewing or developing the assessment items/tasks and marking schemes/rubrics for the project (NTS 3k-3q).

10 minutes	Content Exploration

4. As a subject group, select one problem or topic/idea/term in the week's content that is challenging (not a routine procedural question, but one that requires conceptual understanding) and then
   - use the app, ask it to solve the problem step by step and explain the reasoning behind each step (you may be assisted by a teacher who is strong in the content area)
   - all teachers in the group work through the problem individually for 3–4 minutes
   - the group then works through the problem together, step by step
   - highlight the most common errors made at each step (NTS 1c, 1e, 2c, 2e, 3e-3j).

Note

1. This part is not 'How do I teach this?' but how do I teach it for learners to understand?
2. If the group still shares the same content gap, use the app: ask it to solve the problem step by step and explain the reasoning behind each step.
3. Problems/ideas discussed should be sourced from the Teacher Manual content, not invented separately. This keeps the content exploration directly connected to what teachers will deliver.

30 minutes	Enactment

5. Enact a teaching activity based on your learning plan (NTS la-1b, 1e-lg, 2c, 2e, 2f, 3c-3j and 3o).
   a. Colleagues observe with specific attention to clarity of concept explanation, use of familiar Ghanaian examples and how misconceptions are handled if they arise.
   b. After the enactment, ask for structured feedback from the group
      i. was the concept explained in a way that a learner with no prior knowledge could follow?
      ii. were the examples grounded in contexts familiar to Ghanaian learners?
      iii. did any misconceptions surface and if so, how were they addressed?

5 minutes	Reflection

6. Reflect and share your views on the session. One thing that worked well, what did not work well and what you will do differently. (NTS 1a, 1b).
7. Identify a critical friend to observe and provide feedback on your lesson in relation to the PLC Session conducted (NTS 1f, 1g).
8. Remember to

- transfer your chats into your learning planner template.
- read your Teacher Manual and related Learner Material in preparation for the next session. Take note of the recap weeks, which are linked to Years 1 and 2 (NTS 3a).
- integrate National Values, GESI, SEL, and 21st century skills into your learning activities, assessment tasks, and rubrics/mark schemes.

WEEK 3

Learning Indicators

1. *Prepare technical reports including flowcharts or pseudo codes that outline the sequence of steps to solve problems*
2. *Peer-review and critique the technical reports*

FOCAL AREA: REPORTING ROBOT SOLUTIONS TO IDENTIFIED PROBLEMS

Introduction

Following prior work on analysing local problems and defining a robot's role, this lesson focuses on detailing the step-by-step operation of a potential robotic solution. Learners outline these sequences using flowcharts or pseudocode as part of a technical report. Emphasis is placed on preparing clear documentation and presenting findings. Additionally, skills in peer-reviewing and constructively critiquing these technical reports are developed, fostering improved technical communication.

Building on Foundational Concepts

This lesson builds upon prior learning regarding robot logic representation and capabilities. Foundation in algorithms, flowcharts, and pseudocode from Years 1 and 2 is applied to outline robot operational sequences. Understanding of robot subsystems and mechanics, gained in previous years, informs the breakdown of tasks into steps. Experience with technical reporting and troubleshooting from Year 2 provides the basis for integrating sequence outlines into reports and developing critique skills for improving technical communication.

Recap Weeks

Year 1	**Week 3** • Describe the attributes and functionalities of a robot's subsystems and how they interconnect **Week 5** • Evaluate the use of logic and loop diagrams and demonstrate their use in control systems' design. **Week 17** • Determine the Inputs, processes, and outputs required to solve a particular problem. • Define solutions to basic automated and robotic problems using algorithms, pseudocodes, and flowchart diagrams. **Weeks 19 & 20** • Create robots using robotic kits and/or local materials to implement basic mechanics for actuation (e.g., Gripping Figures, Lifting Mechanisms).

Year 2	**Week 1** • Write publishable articles on topics related to ethics, safety and robot coexistence in society. **Week 3** • Utilise component and system diagrams to design feedback system for implementing controls in given scenarios. **Week 17** • Define solutions for control and feedback systems using algorithms, pseudocodes, and flowchart diagrams. **Weeks 19 & 20** • Create robots using robotic kits and/or local materials to implement basic mechanics for actuation (e.g., Swinging Mechanisms, Automatic Doors, raking up or out). **Week 23 & 24** • Observe and identify flaws in an operational robot and trace them to either algorithm flaws, design flaws or coding errors. • Use a series of iterations to fix the identified flaws or improve the performance of solutions.

KNOWLEDGE CONTENT

The Importance of Outlining Robot Tasks

For a robot to perform a task effectively, a clear, unambiguous set of instructions outlining the sequence of actions is required. Without careful planning of the logic and flow, programming can become disorganised, leading to errors and unpredictable behaviour. Outlining the steps serves as a blueprint, detailing precisely what the robot should do and in what order, including how it handles different conditions or repeats actions. This planning step is essential before the actual programming process begins. It ensures the envisioned robotic role (identified in the previous lesson) can be translated into actionable instructions.

Representing Robot Logic: Flowcharts

Flowcharts are visual tools used to represent the sequence of steps, decisions, and processes in a system or algorithm. Learners were introduced to basic flowcharts in Year 2 as a way to plan block-based programs (Refer to Year 2, Section 8). In this context, flowcharts are used to map out the operational sequence of a robot performing a specific task related to solving a real-world problem.

Remember that the key components of flowcharts for robot tasks include:

1. **Terminator (Oval):** Represents the start and end points of the robot's sequence.
2. **Process (Rectangle):** Represents a specific action the robot performs (e.g., Move Forward, Grip Object, Activate Sensor).

3. **Decision (Diamond):** Represents a point where the robot's path depends on a condition (e.g., Is Object Detected? Is Path Clear?). There are typically two exit paths, one for 'Yes'/'True' and one for 'No'/'False'.
4. **Input/Output (Parallelogram):** Represents data entering or leaving the system, often corresponding to sensor readings (Input) or displaying information (Output).
5. **Flow-lines (Arrows):** Connect the symbols and indicate the direction of the flow of control.
6. **Connector (Circle):** This may be a **new addition**, as it was not introduced in Years 1 and 2. It is used to indicate a jump in the process flow. On-page connectors link different points on the same page, while off-page connectors are used to link to a process on another page, especially when the flowchart is too large to fit on a single sheet. A video reference has been provided in the reference section to demonstrate how this component is often used. It may come in handy, especially now that learners are developing detailed flowcharts for their proposed robotic solutions.

Representing Robot Logic: Pseudocode

Recall that Pseudocode is a structured, informal text-based method for describing the steps of an algorithm or sequence. It is not a specific programming language but uses keywords and indentation to make the logic clear and easy for humans to read. Pseudocode sits between plain language and actual programming code, allowing technical ideas to be communicated clearly without getting bogged down in syntax rules. You are free to coin your own keywords as long as they are quite understandable and easily bring to mind the action to be carried. To avoid misconceptions or vague interpretation, use easy-to-understand or recognisable keywords. Pseudocode complements flowcharts by providing a more textual description of each step and condition.

Common pseudocode elements for outlining robot tasks include:

1. **Sequence:** Steps are listed one after another in order of execution.

```
•    Step 1
•    Step 2
•    Step 3
```

2. **Input/Output:** Commands to receive information or provide results.

```
•    GET sensor_reading
•    DISPLAY message
```

3. **Decisions:** Represented using keywords such as IF, THEN, ELSE, ELSE IF structures.

```
IF obstacle_detected THEN
   Turn Left
ELSE
   Move Forward
END IF
```

4. **Loops:** Represented using keywords such as WHILE, REPEAT, FOR structures.

```
WHILE battery_level_high DO
   Perform Task
END WHILE
REPEAT 5 TIMES
   Grip and Lift Item
END REPEAT
```

5. **Commands:** Clear, simple instructions related to robot actions (e.g., MOVE FORWARD, TURN LEFT, STOP MOTORS, READ SENSOR).

Translating a Robot Task into Representations

The process of outlining a robot's operational sequence involves:

1. **Defining the Task:** Clearly understanding the specific task the robot needs to accomplish (as identified in the previous lesson).
2. **Breaking Down Steps:** Dividing the task into a series of smaller, manageable, sequential actions. Think about what the robot needs to sense, process, and act upon at each stage.
3. **Adding Decisions and Loops:** Identify points where the robot needs to make a choice based on sensor input or where actions need to be repeated.
4. **Choosing Representation:** Decide whether to use a flowchart (visual) or pseudocode (textual) or both to represent the sequence.
5. **Creating the Representation:** Draw the flowchart using standard symbols (including connectors where necessary for multi-page diagrams) or write the pseudocode using clear language and appropriate structures. Ensure the representation accurately reflects the steps, decisions, and loops required for the task.

For example, outlining the steps for a robot collecting waste from a gutter might involve steps like Move along gutter, Sense for waste, IF waste detected THEN Stop, Activate gripper, Grip waste, Lift waste, Move to collection bin, Release waste, Move back to gutter END IF, Repeat until end of gutter. This sequence would then be translated into a flowchart or pseudocode.

The flowchart for this task may look like what is portrayed in Figure 3.1, pay particular attention to how connectors C1 and C2 are being used.:

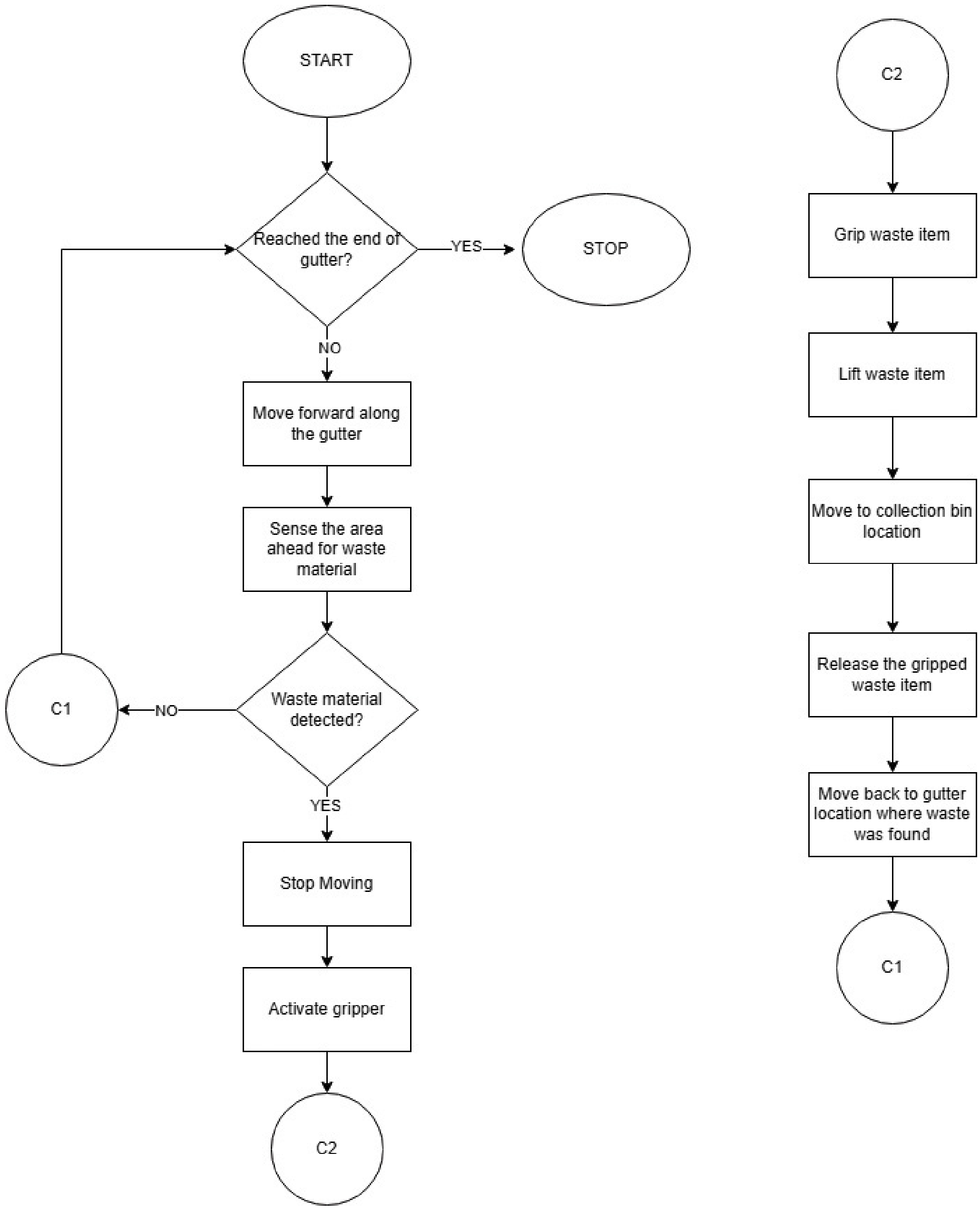


Figure 3.1: *A flowchart for a robot collecting waste from a gutter*

The pseudocode for this task may look like this:

```
START
  REPEAT UNTIL robot is at the end of the gutter
    Move forward along the gutter
    Sense the area ahead for waste material
    IF waste material is detected THEN
      Stop the robot's movement
      Activate the gripper mechanism
      Grip the detected waste item
      Lift the waste item
```

```
        Move to the designated collection bin location
        Release the gripped waste item
        Move back to the gutter location where waste was found
      END IF
    END REPEAT
STOP
```

Choosing Your Planning Tool

Deciding whether to create the flowchart or write the pseudocode first often depends on personal preference and the complexity of the task.

1. **Flowcharts** offer a visual representation of the logic and flow of the robot's actions. For some learners, especially those who are visual thinkers, starting with a flowchart can make it easier to map out the sequence, identify decision points, and visualise loops before getting into the details of textual instructions. This can help ensure the overall logic is sound.
2. **Pseudocode** provides a more structured, text-based outline that is closer to actual programming code. For learners who prefer working with text or who are already comfortable with basic programming concepts, starting with pseudocode can be straightforward. It allows for a detailed description of each step using simple commands, which can then be directly translated into code or a flowchart.

Neither approach is universally "better," but starting with the flowchart might be considered easier for many beginners or when dealing with complex decision paths, as the visual layout can simplify the planning process. Conversely, starting with pseudocode might feel easier for those more comfortable with coding or when the sequence is primarily linear with fewer complex branches. Often, people move between both representations as they refine their plan.

Integrating Representations into a Technical Report

The flowchart or pseudocode outlining the robot's operational sequence is a key component of a technical report that describes the envisioned robotic solution. It provides the detailed "*how*" behind the "*what*" (the robot's defined role). This representation should be included in the adapted report structure to clearly illustrate the robot's planned actions.

The flowchart or pseudocode should be included as a dedicated section immediately following the description of the "*Proposed Robotic Role*." A suitable title for this section/chapter is "Operational Sequence Outline". This chapter will contain the flowchart or pseudocode that outlines the step-by-step actions the robot would perform to execute the proposed role.

Note the following:

1. Learners should choose either a flowchart or pseudocode representation, or they may provide both if it enhances clarity.

2. Ensure the representation accurately reflects the task defined in the "Proposed Robotic Role" section.
3. If using a flowchart that spans multiple pages, ensure on-page and off-page connectors are used correctly.
4. Reference this "Operational Sequence Outline" chapter from within the "Proposed Robotic Role" chapter (Chapter 6) to direct the reader to the detailed steps.
5. Alternatively, if the representation is simple and fits on one page without disrupting the report's flow, it could be included directly within the "Proposed Robotic Role" chapter as a figure or embedded text, clearly labelled. For more complex sequences, the dedicated chapter is recommended for clarity.
6. The complexity and detail of the flowchart or pseudocode should be appropriate for a high-level conceptual design, focusing on the main steps, decisions, and loops.

This placement ensures that the detailed operational plan logically follows the description of the robot's intended function, providing the technical specifics of the proposed role within the structured report format.

Critiquing Technical Reports

Critique is an indispensable part of the engineering and scientific process. Technical reports, which document analysis, designs, and findings, serve as critical communication tools. Evaluating these reports through critique ensures their quality, accuracy, and effectiveness. The essence of critique lies in subjecting technical work to careful examination to identify strengths, weaknesses, and areas for improvement.

They can be carried out personally or in a peer review fashion. When done personally it helps the author to critically review one's work to spot errors in logic, clarity, or completeness before sharing it with others. It allows for self-correction and refinement, leading to a stronger final document. Having colleagues evaluate technical reports in a peer review fashion provides fresh perspectives. Reviewers may identify issues the author overlooked, suggest alternative approaches, or point out areas where the explanation is unclear. Peer review strengthens the work, improves the technical communication skills of both the author and the reviewer, and is a standard practice in professional environments.

Process for Critiquing a Technical Report

A systematic approach to critiquing a technical report helps ensure a thorough and fair evaluation. The following steps outline a common process:

1. **Understand the Purpose and Scope:** Begin by understanding the overall objective of the report and the specific problem and proposed solution it aims to address. Familiarise yourself with the intended audience.
2. **Read Through Entirely:** Read the complete report once to get a general understanding of its content, structure, and flow. Note initial impressions regarding clarity and completeness.

3. **Evaluate Against Criteria:** Read the report again, this time focusing on specific criteria. For a technical report on a proposed robotic role, relevant criteria would include:
   a. Clarity and thoroughness of the problem description and analysis.
   b. Logical flow and justification in linking the problem characteristics to the robot's suitability.
   c. Clarity and specificity of the proposed robotic role.
   d. Accuracy and readability of the operational sequence outline (flowchart or pseudocode) and whether it accurately represents the defined role.
   e. Realism and potential impact of the proposed solution (considering technical feasibility at a high level and societal/ethical implications, drawing on Year 1 and 2 knowledge).
   f. Overall structure, organisation, clarity of language, and professionalism of the report.
4. **Identify Strengths:** Note the aspects of the report that are well-done, clear, insightful, or particularly effective. Recognizing strengths provides positive reinforcement and helps the author understand what to continue doing.
5. **Identify Areas for Improvement:** Pinpoint sections or aspects of the report that are unclear, inaccurate, incomplete, illogical, or could be strengthened. Be specific about the issues found.
6. **Formulate Constructive Feedback:** Based on the identified strengths and areas for improvement, prepare comments and suggestions for the author.

Providing Constructive Critique

The goal of critique is to help the author improve their work. Therefore, feedback should be constructive, specific, actionable, and delivered in a supportive manner.

Key elements of constructive critique include:

1. **Focus on the Work, Not the Person:** Comments should address the content, structure, and technical aspects of the report, avoiding personal remarks.
2. **Be Specific:** General comments like "This is unclear" are not helpful. Instead, point to the specific sentence, paragraph, diagram, or section that is unclear and explain why it is unclear (e.g., "In the 'Problem Context' section, the description of the current process is unclear on page 3 because it does not explain who performs the task currently").
3. **Provide Actionable Suggestions:** Where possible, suggest how an area for improvement could be addressed. Instead of saying "This flowchart is confusing," suggest "The flowchart could be clearer if the decision points were labelled with specific conditions" or "Consider using an on-page connector to simplify the flow on this page."
4. **Be Balanced:** Include comments on the strengths of the report as well as areas for improvement. This provides perspective and encourages the author.
5. **Use Neutral and Professional Language:** Phrase comments respectfully and objectively. Avoid overly negative or critical language.

6. **Justify Comments:** Explain the reasoning behind your critique. Why do you think something is a strength or a weakness? Referencing the evaluation criteria helps justify your points.

Learning Tasks

These tasks require learners to outline the sequential steps for the robotic role they defined in their previous technical report, represent these steps using flowcharts or pseudocode, and integrate this representation into their existing report. Depending on the available time or resources, administer one or more of the following learning tasks to help learners reinforce understanding and acquire new knowledge or skills.

1. Refer to the specific robotic role or task that you defined for your chosen local problem in the technical report prepared in the previous section. Break down this defined robotic role into a clear sequence of simple, ordered steps that the robot would need to perform to execute that role.

   Ensure the steps are specific actions or conditions relevant to a robot's capabilities, directly aligned with the defined role. Consider any conditions or points where the sequence might change based on the problem context.

2. Translate the sequence of steps identified for your specific robotic role in Learning task 1 into a flowchart.

   Use standard flowchart symbols to represent the actions, decisions, and logic flow of your robot's operational sequence. Ensure the flowchart logically follows the sequence of steps identified in Learning task 1.

3. Translate the sequence of steps identified for your specific robotic role in Learning task 1 into pseudocode.

   Use clear, simple text and appropriate keywords (such as IF...THEN...ELSE, REPEAT, WHILE) to describe the sequence of actions, decisions, and any loops involved in the robot's operation. Structure the pseudocode using indentation to clearly show the logic flow.

4. Integrate the sequence of steps (from Learning task 1) and either the flowchart (from Learning task 2) or the pseudocode (from Learning task 3) into the existing technical report you prepared in the previous section by following these steps:

   a. Create a dedicated section in your report titled **Operational Sequence Outline**.

   b. Include the sequence of steps and your chosen representation (flowchart or pseudocode) within this new chapter.

   c. Ensure the operational sequence accurately reflects the robotic role described in the **Proposed Robotic Role** section/chapter.

   d. Make a reference to the **Operational Sequence Outline** section chapter from within the **Proposed Robotic Role**, indicating that the detailed operational steps are provided in the subsequent section.

e. Update your Table of Contents to include the new section/chapter.

5. Exchange the technical report you have prepared with one or more of your classmates. Carefully read and analyse the report you receive by following these steps:

   a. Evaluate the report based on criteria such as the clarity of the problem analysis, the logic and suitability of the proposed robotic role, the accuracy and readability of the operational sequence outline (flowchart or pseudocode), and the overall structure and clarity of the document.

   b. Prepare written feedback for the author of the report. In your feedback, clearly identify the strengths of the report. Also, identify specific areas where the report could be improved, providing concrete and constructive suggestions for making those improvements. Ensure your feedback is respectful and focuses on the work itself.

PEDAGOGICAL EXEMPLARS

Consider the shared characteristics and individual differences among learners, including their interests, readiness, and learning styles. Plan instructional approaches that address these variations. Offer multiple ways for learners to engage with the content and demonstrate their understanding, ensuring all learners can achieve the stated learning outcomes.

1. **Step 1: Recap and Transition to Outlining Operations**

   Begin by briefly reviewing the previous lesson's focus on analysing a specific local problem, defining a robot's role, and documenting this in a report. Introduce the activities for this lesson, highlighting the shift to detailing how the robot performs its defined role by outlining the step-by-step operational sequence using specific technical tools.

To ensure differentiation and inclusivity

   a. Use visual prompts related to the previous lesson's technical reports.

   b. Clearly state the learning objectives for the current lesson, providing them verbally and in written format.

2. **Step 2: Guiding the Breakdown of the Robot Role (Learning Task 1)**

   Direct learners to refer to the specific robotic role they defined in their previous technical report. Guide them through the process of breaking down this defined role into a clear sequence of simple, ordered steps. Facilitate brainstorming or provide examples of how complex actions are broken into simpler ones.

To ensure differentiation

   a. Provide structured worksheets or sentence starters to help learners break down their specific robot role into sequential steps.

b. Work closely with learners who find it challenging to translate the role into discrete actions.

c. Challenge learners with more complex roles to consider sub-sequences or parallel actions.

3. **Step 3: Facilitating Flowchart and Pseudocode Representation (Learning Tasks 2 & 3)**

   Instruct learners on the use of flowcharts and pseudocode as tools for representing the operational sequence identified in Task 1.1. Briefly recap standard flowchart symbols (referencing Year 2 knowledge and introducing the connector symbol if new) and basic pseudocode structures and keywords. Guide learners as they translate their sequence into either a flowchart or pseudocode.

To ensure differentiation

a. Provide visual aids showing standard flowchart symbols and common pseudocode keywords with examples.

b. Offer templates for simple flowcharts or pseudocode structures.

c. Allow learners to work in well-balanced pairs.

d. Provide one-on-one support for learners struggling with the logic or syntax.

e. Ensure visual accessibility for flowcharts; verbal descriptions or digital tools might be helpful.

4. **Step 4: Supporting Integration into the Technical Report (Learning Task 4)**

   Guide learners through the process of integrating the sequence of steps and their chosen representation (flowchart or pseudocode) into the technical report they began in the previous lesson. Explain where this new content should be placed within the adapted report structure (i.e., the Operational Sequence Outline). Support learners in updating their existing reports and Table of Contents.

To ensure differentiation and inclusivity

a. Provide a clear outline or checklist for the updated report structure.

b. Offer support with formatting and incorporating diagrams digitally or manually.

c. Allow flexibility in the presentation format if a standard report structure is a barrier.

5. **Facilitating Peer Review and Critique (Learning Task 5)**

   Establish a process for learners to exchange their technical reports for peer review. Instruct learners on the purpose of critique in technical fields and the principles of providing constructive feedback (referencing knowledge content on critiquing). Guide them in evaluating their peer's report based on defined criteria and formulating specific, helpful written comments addressing both strengths and areas for improvement in the problem analysis, proposed role, and operational sequence outline.

To ensure differentiation and inclusivity

a. Provide a structured feedback form based on the evaluation criteria.
b. Model how to give constructive criticism using example reports (anonymised).
c. Work with learners who may find it difficult to articulate feedback in writing by allowing verbal discussion first.
d. Ensure all learners have equal access to the reports being reviewed (e.g., clear copies, digital format).

6. **Step 6: Discussion and Reflection**

   Facilitate a class discussion following the peer review activity. Discuss what was learned through giving and receiving critique. Highlight common strengths and areas for improvement observed in the reports. Encourage reflection on how documenting and critiquing operational sequences contributes to developing clearer technical ideas.

KEY ASSESSMENT LEVEL 4: ESSAY

1. **Instructions:** Answer the following questions with clear explanations and apply concepts learned in the lesson.

   a. Describe a simple robot task (different from the waste collection example) that involves both sequential steps and at least one decision point. Translate the steps for this task into either a flowchart or pseudocode. (6 marks)

   b. Explain why including the operational sequence outline (via flowchart or pseudocode) is an important component in the technical report describing a proposed robotic solution. How does it enhance the communication of the idea? (4 marks)

   c. You are peer-reviewing a report where the proposed robotic role seems suitable for the problem, but the pseudocode outlining the operational sequence appears illogical for a specific part of the task. Describe how you would formulate constructive feedback for the author regarding this issue, applying principles of constructive critique. (5 marks)

   d. **Individual** Practical Task - (Suggested Modes: Individual Project Assessment)

Project Title: Operational Sequence Analysis and Critique

Task: You will be provided with a description of a moderately complex robot task scenario for a specific problem and either a flowchart or pseudocode representing its operational sequence.

1. Study the provided robot task scenario and the corresponding flowchart or pseudocode. Evaluate whether the representation accurately and logically outlines the steps needed for the robot to perform the task as described.
2. Identify the strengths of the provided representation (e.g., clarity, logical flow, correct use of symbols/structure). Identify any weaknesses or potential flaws in the logic or representation (e.g., missing steps, incorrect decision paths, unclear loops, incorrect symbol usage if a flowchart).

3. Based on your critique, suggest specific improvements to the flowchart or pseudocode to make it more accurate, complete, and logical for performing the task described in the scenario. If the representation has significant flaws, propose a corrected version.
4. Prepare a short report or presentation detailing your analysis, identified strengths and weaknesses, and proposed improvements. Justify your critique and suggested changes by referencing the robot task scenario and principles of flowchart/pseudocode representation.

(***15 marks***)

Conclusion

In this lesson, the focus was placed on detailing how an envisioned robot performs its task. Operational sequences were outlined using flowcharts and pseudocode, and these representations were integrated into the technical report structure. The process of peer-reviewing and critiquing reports was undertaken. This work emphasised the importance of clear documentation and constructive feedback for improving technical communication and refining proposed robotic solutions.

PLC Session 3

45 minutes	Planning & Reviewing the Learning Plan

1. Begin the session by recapping lessons from the previous year/session/week, focusing on how learning was applied.
2. Use your subject specific app to explore the content before planning. Ask the app to
   a. explain the week's concepts in a practical way that will make it easier for learners to understand
   b. identify two or three common misconceptions learners may have on the week's content and how to address them using contexts familiar to Ghanaian learners. (NTS 3a, 3g and 3m).
3. Use your subject specific app to plan your lesson. Ask the app to
   a. suggest the most effective method for teaching the week's concepts to learners with diverse backgrounds.
   b. suggest learning activities, differentiation and essential questions based on the method of teaching (NTS 3a-3q)
   c. review your prepared learning plan and address any challenging issues you may have identified in your subject area (NTS 3a-3c).

Note

1. Take note of the recap topics linked to Years 1 and 2 while preparing for your lesson.
2. Refer to the Values Learning Community Handbook and link the values for the week in your lesson to ensure that learners are applying what they learn.

3. Plan the week's key assessment with your app
4. The key assessment for the week is ***essay***. Chat with your app to plan this assessment, indicating what you and your learners will do before, during and after the essay (NTS 3k-3q).

10 minutes	Content Exploration

4. As a subject group, select one problem or idea/term in the week's content that is challenging (not a routine procedural question, but one that requires conceptual understanding) and then
   a. use the app, ask it to solve the problem step by step and explain the reasoning behind each step (you may be assisted by a teacher who is strong in the content area)
   b. all teachers in the group work through a similar problem individually for 3–4 minutes
   c. the group then works through the problem together step by step
   d. highlight the most common errors made at each step and discuss (NTS 1c, 1e, 2c, 2e, 3e-3j).

Note

- This part is not just about 'How do I teach this?' but how do I teach it for learners to understand?
- If the group still shares the same content gap, use the app: ask it to solve the problem step by step and explain the reasoning behind each step.
- Problems/ideas discussed should be sourced from the Teacher Manual content, not invented separately. This keeps the content exploration directly connected to what teachers will deliver.

30 minutes	Enactment

5. Enact a teaching activity based on your learning plan (NTS la-1b, 1e-lg, 2c, 2e, 2f, 3c-3j and 3o).
   a. Colleagues observe with specific attention to clarity of concept explanation, use of familiar Ghanaian examples and how misconceptions are handled if they arise.
   b. After the enactment, ask for structured feedback from the group
      i. was the concept explained in a way that a learner with no prior knowledge could follow?
      ii. were the examples grounded in contexts familiar to Ghanaian learners?
      iii. did any misconceptions surface and if so, how were they addressed?

5 minutes	Reflection

6. Reflect and share your views on the session. One thing that worked well, what did not work well and what you will do differently. (NTS 1a, 1b).

7. Identify a critical friend to observe and provide feedback on your lesson in relation to the PLC Session conducted (NTS 1f, 1g).

8. Remember to

a. transfer your chats into your learning planner template.

b. read your Teacher Manual and related Learner Material in preparation for the next session. Take note of the recap weeks, which are linked to Years 1 and 2 (NTS 3a).

c. integrate National Values, GESI, SEL, and 21st century skills into your learning activities, assessment tasks, and rubrics/mark schemes.

Section Review

Progression through key stages in applying robotics to real-world issues is presented in this section. Initially, focus was placed on identifying problems suited for robotic assistance based on capability alignment. Subsequently, the analysis of specific local challenges was undertaken, leading to the definition and documentation of precise robotic roles. Finally, attention was given to detailing the robot's operational steps using flowcharts and pseudocode, integrating these into reports, and critiquing the documented solutions for improved communication and refinement.

Additional Reading

Resource	QR Code to Accessing Resource
1. Tutorial on using connectors in Flowcharts (Video Resource).	
2. How to Review and Provide Constructive Feedback as a Peer Reviewer (Paper Publication)	
3. Sample Spike Prime Projects (Video Resource)	
4. 10 Incredible LEGO SPIKE Prime robots for Medical Applications (Video Resource)	

MARKING SCHEMES FOR THE KEY ASSESSMENTS

WEEK 1

QUESTION 1

Marking Scheme - Short Answer and Extended Response Essays

Question 1 – 4 marks	
Description of local problem	**1 mark** – Specific description of a local industry's problem
Explanation and Justification	**2 marks –** Links the problem to at least two relevant characteristics like Dangerous Environments, Repetitive Tasks, Inaccessible Locations, etc., and links these characteristics to corresponding robotic capabilities like Movement, Manipulation, Sensing, etc.
Relevance	**1 mark** – Logical reasoning for why a robot fits this specific problem.
Question 2 – 3 marks	
Identification of specific robotic task:	**1 mark** – Identifies a relevant task, e.g., 'collecting plastic using a net', 'navigating to waste patches'.
Justification based on ocean challenges	**2 marks** – Explains why robot is better than human carrying out this task due to factors like endurance for long periods, ability to operate in harsh conditions/currents, working in remote ocean areas.
Question 3 – 3 marks	
Explanation of Sensing + Continuous Operation:	**2 marks –** Explains how sensing allows data collection and how continuous operation allows for long-term monitoring without breaks.
Specific Example:	**1 mark** – Provides a relevant example like monitoring water temperature/pollution in a river, or air quality in a factory.

QUESTION 2

Robotic Intervention for Radio Mast Inspection (Total: 12 marks)

Assessment Area	Criteria	Marks
1. Explanation of Risks and Logistical Issues (Human-led Inspection)	• Clear identification of basic risks (e.g., height, safety) • Recognition of logistical complexity (e.g., rural access)	**1 mark**
	• Detailed explanation of multiple risk dimensions (e.g. fall hazard, extreme weather) • Some insight into human resource or time implications	**2 marks**
	• Sophisticated breakdown of risk types (e.g., health, systemic inefficiency) • Integration of contextual factors like terrain, technician fatigue, regulatory compliance	**3 marks**

2. Evaluation of Robotic Capabilities vs Human Methods	• General mention of robotic traits (e.g., speed, sensors) • Basic contrast with humans	**1 mark**
	• Thoughtful comparison of at least two capabilities (e.g., drone mobility vs climbing access, sensing accuracy vs visual fatigue) • Some link to operational improvement	**2 marks**
	• Sophisticated evaluation with clear examples and technical reasoning (e.g., GPS-enabled drones reduce inspection time by 80%) • Well-explained limitations of human inspections	**3 marks**
3. Assessment of Limitations or Trade-offs	• States that robots have limitations (e.g., cost, maintenance) • Limited exploration	**1 mark**
	• Discusses at least one plausible downside (e.g., initial investment, data security) • Explores impact on existing workflows	**2 marks**
	• Detailed assessment of multiple limitations • Weighs ethical, social, or economic consequences (e.g., job displacement, tech dependency)	**3 marks**
4. Justified Conclusion on Sustainability and Ethics	• States that robotics is helpful or preferable • Little justification	**1 mark**
	• Provides reasonable justification based on robotic benefits • Mentions broader context (e.g., community benefit, long-term safety)	**2 marks**
	• Balanced conclusion that integrates evidence from evaluation • Addresses sustainability, ethics, and context relevance thoroughly	**3 marks**

QUESTION 3

Rubric - Group-Based Task/Project

Criteria	Mark Distribution	
Problem Description & Analysis **(6 marks)**	**1-2 Marks**	Problem is vaguely described. Analysis of characteristics is missing or unclear.
	3-4 Marks	Problem is described. Identifies some relevant characteristics, but analysis is superficial.
	5-6 Marks	Problem is clearly and specifically described with impact. Critical analysis identifies key characteristics making it suitable for robotics set against what areas/functions would be best not automated.

Proposed Robotic Role **(4 marks)**	**1 Mark**	Robotic role is unclear or not relevant to the problem.
	2-3 Marks	A robotic role is suggested, but it is general or not well-linked to the problem.
	4 Marks	A specific, relevant robotic role/task is clearly proposed for the problem.
Justification of Suitability **(6 marks**	**1-2 Marks**	Justification is absent or weak, without linking to robot characteristics/capabilities.
	3-4 Marks	Attempts to justify suitability, but links to characteristics/capabilities are vague or incomplete.
	5-6 Marks	Provides a clear and logical justification for the robotic role, explicitly linking it to relevant robot characteristics and capabilities discussed in class.
Comparison to Current Method **(2 marks)**	**1 Mark**	Mentions the current method but does not clearly explain advantages of the robot.
	2 Marks	Clearly outlines the advantages of the proposed robotic approach compared to the current method.
Report/Presentation Clarity **(2 marks)**	**1 Mark**	Report/presentation is understandable but could be better structured or clearer.
	2 Marks	Report is well-structured, clear, and easy to follow OR Presentation is clear, organised, and well-articulated.
Total	**20 Marks**	

WEEK 2

QUESTION 1 (GROUP-BASED TASK)

Rubric - Group-Based Task/Project

Criteria	Mark Distribution	
Problem Analysis & Suitability **(15 marks)**	**1 - 5 Marks**	Problem is vaguely defined, or analysis is superficial. Suitability justification is weak, lacking links to characteristics/capabilities. Limited understanding of context/impact.
	6 - 10 Marks	Problem is defined, some analysis provided. Suitability justification attempts links but lacks depth or clarity. Basic understanding of context/impact shown.
	11 - 15 Marks	Problem is clearly defined and analysed in depth (causes, impact, context). Suitability for a robot is strongly justified with clear, specific links between problem characteristics and robotic capabilities/ principles.

Proposed Robotic Role & Concept **(8 marks)**	**0 - 2 Marks**	Robotic role is unclear or inappropriate. Conceptual system components/operation are missing or irrelevant.
	3 - 5 Marks	Robotic role is suggested but lacks specificity. Conceptual system is basic, some relevant components mentioned. Basic operational idea.
	6 - 8 Marks	A specific, well-defined robotic role is proposed. Conceptual system description includes relevant components (sensors, actuators) and a clear high-level idea of how they function together to perform the role.
Impact, Benefits & Challenges **(5 marks)**	**0 - 1 Mark**	Impact/benefits not discussed or are unclear. Challenges are not mentioned or are superficial
	2 - 3 Marks	Mentions some potential impacts/benefits. Briefly discusses one or two obvious challenges.
	4 - 5 Marks	Clearly explains how the robotic role improves efficiency and addresses impacts. Discusses significant potential challenges in implementing the prototype in the local context.
Report & Presentation **(2 marks)**	**0 Marks**	Report structure is poor. Presentation is disorganised. Communication is unclear.
	1 Mark	Report follows basic structure but lacks detail/clarity. Presentation is understandable but could be better organised.
	2 Marks	Report follows a coherent structure, is well-organised, clear, and detailed. Presentation is clear, well-structured, and effectively communicates analysis and proposal.
Total	**30 Marks**	

QUESTION 2

Rubric - Individual-Practical Task

Criteria	Mark Distribution	
Problem Analysis & Suitability **(10 marks)**	**1 - 3 Marks**	Problem is vague or lacks analysis. Suitability justification is weak or missing links to characteristics/capabilities. Limited understanding of context/impact.
	4 - 7 Marks	Problem is defined, some analysis provided. Suitability justification attempts links but lacks depth or clarity. Basic understanding of context/impact shown.
	8 - 10 Marks	Problem is clearly defined and analysed in depth (causes, impact, context). Suitability for a robot is strongly justified with clear, specific links between problem characteristics and robotic capabilities/principles.

Conceptual Design & Feedback **(8 marks)**	**1 - 2 Marks**	Conceptual system is unclear. Components not identified or irrelevant. Feedback loop not mentioned or incorrectly explained.
	3 - 5 Marks	Conceptual system is basic, some components mentioned. Feedback loop mentioned but not clearly explained how it enables the task.
	6 - 8 Marks	Clear description of the conceptual system and robot task. Identifies relevant sensor/actuator types. Explains clearly how feedback is used for autonomous task performance.
Impact, Benefits & Challenges **(5 marks)**	**0 - 1 Mark**	Impact/benefits not discussed or are unclear. Challenges are not mentioned or are superficial
	2 - 3 Marks	Mentions some potential impacts/benefits. Briefly discusses one or two obvious challenges.
	4 - 5 Marks	Clearly explains how the robotic role addresses the problem's impact and provides benefits. Discusses significant potential challenges in implementing the system in the real community.
Report & Presentation **(2 marks)**	**0 Marks**	Report structure is poor. Presentation is disorganised. Communication is unclear.
	1 Mark	Report follows basic structure but lacks detail/clarity. Presentation is understandable but could be better organised.
	2 Marks	Report follows a coherent structure, is well-organised, clear, and detailed. Presentation is clear, well-structured, and effectively communicates analysis and proposal.
Total	**25 Marks**	

WEEK 3 – Group Task

QUESTION 1

Marking Scheme - Short Answer and Extended Response Essays :

Question 1 - 6 marks	
Task Description	**1 mark** - Clear and simple task involving sequence and decision
Translation with accurate symbols/ keywords	**3 marks** - correct use of symbols/keywords/structure
Translation correctly the stated logic	**2 marks** - accurately reflecting the task logic
Question 2 - 4 marks	
Explanation of Importance	**2 marks** - Explains that it provides a detailed plan, shows how the robot performs the role, serves as a blueprint before programming

Enhancement of Communication	**2 marks -** Explains how it clarifies the idea for the reader, makes the proposal more concrete, helps identify potential flaws in the logic
Question 3 - 5 marks	
Identification of issue	**1 mark -** correct identification the specific issue location
Explanation of logic	**2 marks -** correct explanation of why the logic is illogical in that part
Constructive feedback	**2 marks** - formulating specific feedback, actionable, balanced (mentioning suitability of role), and uses neutral language

QUESTION 2

Rubric - Individual-Practical Task

Criteria	Mark Distribution	
Analysis of Representation **(9 marks)**	**1 - 3 Marks**	Analysis is superficial. Identifies few or no strengths/ weaknesses in the representation or logic. Does not accurately link representation to scenario.
	4 - 6 Marks	Identifies some strengths/weaknesses but lacks depth or specific examples. Attempts to link representation to scenario with partial success.
	7 - 9 Marks	Accurately analyses the representation (flowchart or pseudocode) against the task scenario. Clearly identifies specific strengths and weaknesses in logic, accuracy, and structure with justifications.
Proposed Improvements **(4 marks)**	**0 - 1 Marks**	Suggestions for improvement are missing, unclear, or do not address the identified weaknesses.
	2 - 3 Marks	Suggestions are general or only address minor issues. Proposed corrected version (if applicable) has flaws.
	4 Marks	Proposes specific, logical, and constructive improvements that directly address identified weaknesses. Proposed corrected version (if applicable) is largely accurate and improved.
Documentation/ Presentation **(2 marks)**	**0 Marks**	Report/presentation is poorly structured or unclear. Communication is difficult to follow.
	1 Mark	Report/presentation is understandable but could be better organised or clearer in explaining analysis/proposals.
	2 Marks	Report is well-structured, clear, and effectively communicates analysis and proposed improvements with justifications. Presentation (if applicable) is clear and well-articulated.
Total	**15 Marks**	

SECTION 2: ROBOT CONTROL PRINCIPLES 3

STRAND: PRINCIPLES OF ROBOTIC SYSTEMS

Sub-Strand: Robot Control Principles

Week 4

Learning Outcome: *Design and implement feedback-driven autonomous systems to improve efficiency in an identifiable local industry*

Content Standard: *Develop knowledge of designing feedback systems for autonomous control of prototype robotic systems.*

Weeks 5 & 6

Learning Outcome: *Demonstrate skills in reverse-engineering controls of functional autonomous robotic systems through observation and analysis*

Content Standard: *Demonstrate skills in reverse engineering of controls in robotic systems.*

Week 7 & 8

Learning Outcome: *Create test plans for testing robotic systems to uncover faults and go through an iterative fault detection and correction process for defect fixing in robots.*

Content Standard: *Demonstrate practical skills in evaluating and improving existing robotic systems.*

INTRODUCTION AND SECTION SUMMARY

This section is dedicated to advancing learners' proficiency in autonomous robotic systems through a structured progression of design, analysis, and troubleshooting. Focus is placed on applying feedback control principles to create efficient prototypes for local Ghanaian industries, reverse-engineering existing robotic systems to understand their control mechanisms and systematically diagnosing and refining operational robots. Through practical engagement, observation, and documentation, learners are equipped with skills to address real-world challenges, fostering analytical thinking and innovative solutions tailored to community and industrial needs, grounded in prior knowledge and methodical processes.

The learning indicators covered by the section in the following weeks are

Week 4

- Design a prototype feedback-driven autonomous system for efficiency improvement in an identified local industry.

Weeks 5 & 6

- Observe and reverse-engineer the controls of a functional autonomous robotic system.

Weeks 7 & 8

- Test operational robots for specified tasks and observe/document identified flaws.
- Identify faults in robotic systems and trace them to algorithm flaws, design flaws or coding errors.
- Go through series of iterations in fixing the identified flaws or improve robot's performance.

SUMMARY OF PEDAGOGICAL EXEMPLARS

The pedagogical exemplars in this section provide structured guidance for delivering lessons on designing, reverse-engineering, and troubleshooting autonomous robotic systems. Learners are engaged through practical tasks, such as designing feedback-driven prototypes, analysing control mechanisms, and diagnosing flaws, with clear steps for observation, documentation, and iterative refinement. Differentiation is ensured by offering templates, visual aids, and varied expression methods (e.g., verbal, written, diagrammatic), catering for diverse learning needs. Inclusivity is prioritised through accessible resources, peer collaboration, and adaptive tools, fostering analytical skills and practical application in real-world contexts, while emphasising systematic documentation for professional growth.

ASSESSMENT SUMMARY

The following table presents a sample of the key assessments used in this section.

Indicators	DOK Level	Assessment Strategy and/or Mode
1. Design a prototype feedback-driven autonomous system for efficiency improvement in an identified local industry.	**3&4**	Short and Extended Response Essays – (Suggested Modes: Group assignment/homework, Mid-semester Assessment, End-of-semester Assessment)
		Group-Based Task/Project – (Suggested Modes: Group project/research/case study)

1. Observe and reverse-engineer the controls of a functional autonomous robotic system.	**3&4**	Short and Extended Response Essays – (Suggested Modes: Group assignment/homework, Mid-semester Assessment, End-of-semester Assessment)
		Group-Based Task/Project – (Suggested Modes: Group project/research/case study)
1. Test operational robots for specified tasks and observe/ document identified flaws. 2. Identify faults in robotic systems and trace them to algorithm flaws, design flaws or coding errors. 3. Go through series of iterations in fixing the identified flaws or improve robot's performance.	**3&4**	Individual Practical Task – (Suggested Modes: Individual Project Assessment)

WEEK 4

Learning Indicator: *Design a prototype feedback-driven autonomous system for efficiency improvement in an identified local industry*

FOCAL AREA: DESIGNING AUTONOMOUS FEEDBACK SYSTEM PROTOTYPES

Introduction

In this lesson, we will apply our knowledge of feedback control from previous years to design a prototype autonomous robotic system aimed at improving efficiency in a local Ghanaian industry. Efficiency means doing a task well without wasting time, energy, or resources. Many tasks in industries like agriculture, manufacturing, or even local markets can be made more efficient using automation and feedback-driven control. This lesson will focus on designing how such a system would work, outlining its components and logic using our understanding from previous years and our recent skills in documentation.

Building on Foundational Concepts

This lesson builds upon foundational knowledge of robot systems and control. Prior understanding of feedback loops and their contrast with non-feedback systems from Year 1 is applied. The design of feedback systems using component diagrams, covered in Year 2, is central. Knowledge of robot subsystems, sensors, and mechanics from both years informs the selection and integration of components into a prototype concept. Logic diagramming skills are used to outline the autonomous control flow.

Recap Weeks

Year 1	**Week 3** 1. Describe robots and identify the differences between robotic and non-robotic systems. 2. Describe the attributes and functionalities of a robot's subsystems and how they interconnect **Week 4** 1. Contrast non-feedback loop systems and feedback loop systems. **Week 7** 1. Discuss the scientific principles underlying the operation of sensors. **Week 8** 1. Calibrating Linear Sensors for Optimal Performance in Robotic Systems **Weeks 19 & 20** 1. Create robots using robotic kits and/or local materials to implement basic mechanics for actuation.

Year 2	**Week 3** 1. Make use of component diagrams and systems diagrams to design non-feedback systems for implementing controls in given scenarios. 2. Utilise component and system diagrams to design feedback systems for implementing controls in given scenarios. **Week 17** 1. Define solutions for control and feedback systems using algorithms, pseudocodes, and flowchart diagrams. **Weeks 19 & 20** 1. Create robots using robotic kits and/or local materials to implement basic mechanics for actuation.

KNOWLEDGE CONTENT

Autonomous Systems and the Role of Feedback

An autonomous system is one that can operate for extended periods without continuous human intervention. It can make decisions and take actions based on the information it receives from its environment. True autonomy often involves a level of intelligence and adaptability, but at the core, it relies on the system being able to monitor its own performance and external conditions and adjust its behaviour accordingly.

Feedback control is a fundamental principle that enables this autonomy. As studied previously, a feedback system uses the output or state of a system to influence its future behaviour. Information about the current state (measured by sensors) is "fed back" into the control system, which then makes adjustments to the inputs or actions (via actuators) to achieve or maintain a desired state (the setpoint). This continuous monitoring and adjustment loop allows the robot to react to changes and disturbances in its environment without needing explicit step-by-step instructions for every possible situation. For instance, an autonomous sorting system uses a sensor to identify the type of object and feeds this information back to control an actuator that places the object in the correct bin.

Efficiency in Local Industrial Contexts

Efficiency in a local industrial context refers to performing a task with minimal waste of resources – time, materials, energy, and effort. For example, in:

1. **Small-scale Farming:** Inefficiencies might include slow manual planting or harvesting, inconsistent watering leading to crop loss, or time-consuming manual sorting of produce after harvest. Improving efficiency could mean automating precise planting, optimising watering based on soil moisture (feedback), or automating sorting.
2. **Food Processing:** Bottlenecks could be slow mixing, inconsistent heating/cooling, or manual packaging. Efficiency improvement could involve automated mixing, temperature control systems (feedback), or automated packaging lines.

3. **Tailoring/Textile Production:** Inefficiencies might include manual cutting with material waste, or slow and inconsistent stitching. Automation could involve precise automated cutting (though perhaps not fully autonomous with feedback at this level) or feedback systems to ensure consistent stitching speed or tension.
4. **Craft Making:** Repetitive steps like sanding, polishing, or applying finishes could be time-consuming. Automation with feedback could ensure consistent application or finish quality.
5. **Local Transport/Logistics:** Inefficiencies could be in sorting packages, loading/unloading, or navigating tight spaces. Robots with feedback could assist in sorting items or navigating semi-autonomously in defined areas.

Designing for efficiency requires analysing the current process, identifying specific points of inefficiency or waste, and then conceptualising how a robot using feedback could perform that step more effectively.

Designing a Feedback System for Efficiency

The design process for this lesson involves applying the understanding of feedback to a chosen inefficiency:

1. **Problem Analysis:** Deeply understanding the current, inefficient process. What exactly makes it slow, wasteful, or prone to error? (Connects to Year 3, Section 1 problem analysis skills).
2. **Identify Controlled Variable:** What specific aspect of the process needs to be monitored and controlled to improve efficiency? (e.g., the position of a package, the temperature of a drying process, the colour of an item).
3. **Select Sensor:** Which sensor can accurately measure the controlled variable? (Draw on Year 1 knowledge of sensor types – light, colour, touch, ultrasonic, temperature, etc).
4. **Select Actuator:** Which actuator can influence the controlled variable? (Draw on Year 1 knowledge of actuator types – motors, servos, pumps, heaters, etc.).
5. **Define Setpoint:** What is the desired value or state for the controlled variable?
6. **Outline Control Logic:** How will the robot use the information from the sensor, compared to the setpoint, to decide how to activate the actuator? This defines the feedback rule. For a simple system, this might be an "if-then-else" structure.

Sample Scenario/Case Study: Automated Tomato Sorting for Local Markets

The following section presents a scenario in a local context where the problem of inefficiencies can be identified, and suggestions are made to deal with the identified problem from a robotic perspective.

1. **The Efficiency Problem:** Typically, in Africa and in most tomato farming communities in Ghana, after harvesting, tomatoes need to be sorted by ripeness and quality before being sent to market or for further processing. This is often a manual process, just as depicted in ***Figure 4.1.***

Figure 4.1: *Manual tomato sorting in Ghana*

2. **The Efficiency Problem:** Manual sorting of tomatoes is a repetitive, time-consuming, and often inconsistent task. Workers visually inspect each tomato, which can be tiring and lead to errors, especially when dealing with large volumes. Inconsistent sorting results in mixed quality batches, potentially lowering market value and increasing post-harvest losses if riper tomatoes are not sold quickly. The process is also a bottleneck, slowing down the speed at which produce can be prepared for distribution.
3. **Proposed Prototype Robotic System: An Automated Tomato Sorter:** To improve the efficiency and consistency of this sorting process, a simple prototype feedback-driven autonomous system can be designed. This system would automate the task of sorting tomatoes by ripeness based on their colour.
4. **Conceptual Design:** The prototype system would consist of a simple conveyor mechanism, a colour sensor, a small control unit (like a microcontroller), and an actuator to sort the tomatoes into different collection bins. The prototype may have the following components:
   a. **Input/Conveyor:** A simple conveyor belt or a vibrating feeder mechanism to move tomatoes past the sensing area one by one at a controlled speed.
   b. **Sensing:** A colour sensor positioned above or below the conveyor belt. This sensor can detect the colour of the tomato as it passes by. Different stages of ripeness in tomatoes often correspond to different colour ranges (e.g., green for unripe, yellowish red for semi-ripe, deep red for ripe).
   c. **Control:** A small microcontroller or simple logic circuit. This acts as the “brain” of the system, receiving data from the colour sensor and controlling the sorting actuator.
   d. **Actuation:** A simple mechanical **actuator**, such as a small arm, flap, or puff of air, positioned after the colour sensor. This actuator is activated by the controller to divert the tomato off the main conveyor into a designated bin based on its detected colour.
   e. **Power:** A battery or a connection to a local power source to power the components.
5. **Feedback Control Logic:** This system is feedback-driven because the colour sensor provides feedback about the actual state of the tomato (its colour, indicating ripeness). The constituents of this closed feedback system are as follows:

a. **Setpoint:** The system has predefined "setpoints" or target colour ranges corresponding to each ripeness category (e.g., colour values 0-50 for unripe, 51-150 for semi-ripe, 151-255 for ripe, assuming a simple colour value scale).

b. **Sensing:** As a tomato passes the colour sensor, the sensor measures its colour and sends this data (the *actual state*) to the microcontroller (the controller).

c. **Comparison/Control Logic:** The microcontroller compares the sensed colour value to the predefined setpoints. Based on this comparison, the controller decides which category the tomato belongs to.

d. **Actuation:** The controller then sends a signal to the actuator (e.g., a servo motor controlling a flap). If the tomato is detected as "ripe" (e.g., colour value > 150), the controller might activate the actuator to push the tomato into the "ripe" bin. If it is "unripe" (colour < 50), it might be directed to the "unripe" bin, and so on. The actuator's action (output) is determined by the sensor's reading (input/feedback) relative to the setpoint.

6. **Conceptual Design:** The prototype system would consist of a simple conveyor mechanism, a colour sensor, a small control unit (like a microcontroller), and an actuator to sort the tomatoes into different collection bins. The prototype may have the following components:

   a. **Conveyor:** A simple conveyor belt or a vibrating feeder mechanism to move tomatoes past the sensing area one by one at a controlled speed.

   b. **Sensing:** A colour sensor positioned above or below the conveyor belt. This sensor can detect the colour of the tomato as it passes by. Different stages of ripeness in tomatoes often correspond to different colour ranges (e.g., green for unripe, yellowish red for semi-ripe, deep red for ripe).

   c. **Control:** A small microcontroller or simple logic circuit. This acts as the "brain" of the system, receiving data from the colour sensor and controlling the sorting actuator.

   d. **Actuation:** A simple mechanical **actuator**, such as a small arm, flap, or puff of air, positioned after the colour sensor. This actuator is activated by the controller to divert the tomato off the main conveyor into a designated bin based on its detected colour.

   e. **Power:** A battery or a connection to a local power source to power the components.

7. **Feedback Control Logic:** This system is feedback-driven because the colour sensor provides feedback about the actual state of the tomato (its colour, indicating ripeness). The constituents of this closed feedback system are as follows:

   a. **Setpoint:** The system has predefined "setpoints" or target colour ranges corresponding to each ripeness category (e.g., colour values 0-50 for unripe, 51-150 for semi-ripe, 151-255 for ripe, assuming a simple colour value scale).

   b. **Sensing:** As a tomato passes the colour sensor, the sensor measures its colour and sends this data (the *actual state*) to the microcontroller (the controller).

   c. **Comparison/Control Logic:** The microcontroller compares the sensed colour value to the predefined setpoints. Based on this comparison, the controller decides which category the tomato belongs to.

d. **Actuation:** The controller then sends a signal to the actuator (e.g., a servo motor controlling a flap). If the tomato is detected as "ripe" (e.g., colour value > 150), the controller might activate the actuator to push the tomato into the "ripe" bin. If it is "unripe" (colour < 50), it might be directed to the "unripe" bin, and so on. The actuator's action (output) is determined by the sensor's reading (input/feedback) relative to the setpoint.

8. **Pseudocode for Tomato Sorter Robotic System:** The following pseudocode suggests how this can be implemented:

```
START
    //System waits for a tomato
    WAIT FOR TOMATO TO REACH SENSOR AREA

    //Sense the tomato's colour
    READ COLOUR_VALUE FROM COLOUR_SENSOR
    //Make a decision based on colour(Feedback Logic)
    IF COLOUR_VALUE is in RANGE_FOR_RIPE:
        //Activate actuator for ripe bin
        ACTIVATE_ACTUATOR_FOR_RIPE_BIN
    ELSE IF COLOUR_VALUE is in RANGE_FOR_SEMI_RIPE
        //Activate actuator for semi ripe bin
        ACTIVATE_ACTUATOR_FOR_SEMI_RIPE_BIN
    ELSE: //Assume unripe or damaged
        //Activate actuator for other bin
        ACTIVATE_ACTUATOR_FOR_OTHER_BIN
    //Allow tomato to be diverted
    WAIT_FOR_ACTUATOR_ACTION_COMPLETION

    //System is ready for next tomato
    GO_TO_START
END
```

9. **Flowchart for Tomato Sorter Robotic System:** The following flowchart suggests how this can be implemented:

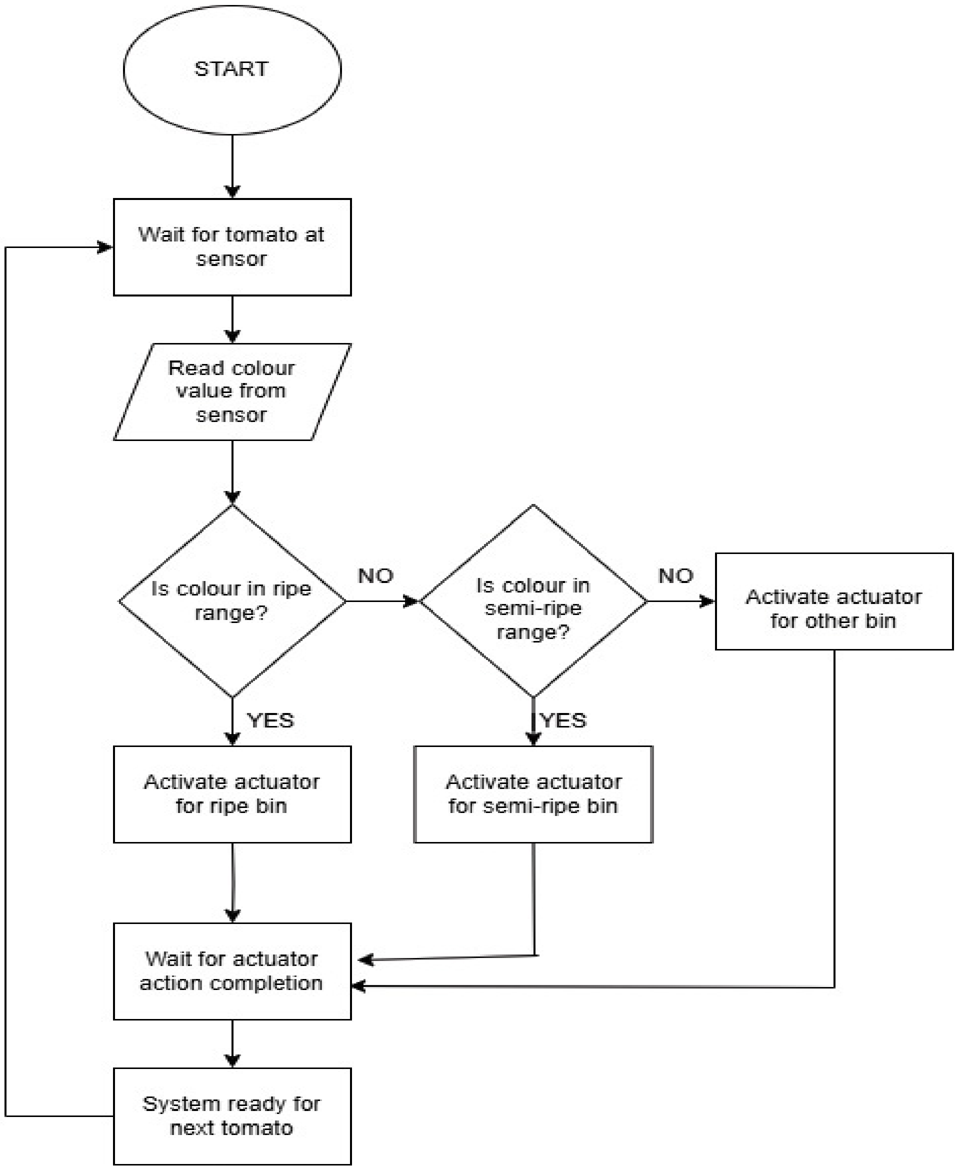


Figure 4.2: *Flowchart of tomato sorting algorithm*

10. **Efficiency Improvement:** Implementing this prototype system would improve efficiency in several ways compared to the manual process:

    a. **Increased Speed and Throughput:** The automated system can sort tomatoes much faster and more consistently than human workers over extended periods without fatigue.

    b. **Improved Consistency:** Sorting is based on objective colour values rather than subjective human judgment, leading to more uniform batches.

    c. **Reduced Labour:** Frees up human workers to perform other tasks that require more complex judgment or dexterity.

    d. **Reduced Post-Harvest Loss:** Faster sorting means less time for ripe tomatoes to spoil before reaching the market. Consistent sorting ensures produce is directed to the appropriate channel quickly.

e. **Operation in Various Conditions:** While manual sorting can be affected by lighting conditions or worker fatigue, a well-calibrated sensor and system can operate consistently.

This sample scenario demonstrates how a relatively simple robotic system, leveraging feedback control based on colour sensing, can address a real inefficiency problem in a local Ghanaian industry, improving both speed and consistency in a crucial post-harvest task.

Learning Tasks

These tasks guide learners through the process of identifying a local industry problem, analysing it for efficiency issues, and designing a feedback-driven prototype system. Depending on the available time or resources, administer one or more of the following learning tasks to help learners reinforce understanding and acquire new knowledge or skills.

1. In small groups, identify a specific task within a local industry (e.g., small-scale farming, food processing, tailoring/textile production, craft making, local transport/logistics or within the school's immediate environment) where efficiency could be significantly improved. Analyse the current process, identify bottlenecks, repetitive tasks, or areas with potential for waste (time, material, energy).
2. For the chosen efficiency problem, design a conceptual or physical prototype feedback-driven autonomous robotic system. This involves:
   a. Defining the specific task that the robot will perform.
   b. Identifying the variable(s) that need to be sensed to control the task autonomously (e.g., position, temperature, colour, level).
   c. Selecting appropriate types of sensors that could measure these variables.
   d. Identifying the actuator(s) the robot would use to make adjustments.
   e. Outlining the feedback control logic – how the sensor input will be used to adjust the actuator output to improve efficiency.
   f. Sketch the system, label components, and describe the feedback loop.
3. In your groups, outline the sequence of steps their feedback-driven prototype system would follow to perform its task efficiently, using either a flowchart or pseudocode (building on previous Year 3 lessons), then document the design in a structured format, including:
   a. Description of the efficiency problem.
   b. Description of the proposed prototype system.
   c. Explanation of the feedback mechanism (controlled variable, sensor, actuator, logic).

d. A flowchart or pseudocode of the operational sequence.

e. Justification of how this feedback system improves efficiency compared to the current method.

In preparing the documentation, ensure that the explanation of the feedback loop and the operational sequence are clear and directly related to solving the efficiency problem.

PEDAGOGICAL EXEMPLARS

This section guides the teacher in facilitating the design and documentation process, focusing on the application of feedback control for efficiency. Recognise that learners possess shared characteristics while also having individual differences in interests, readiness, and learning styles. Plan to address these variations. Provide multiple avenues for learners to engage with the lesson content, adjusting the level of detail or support as needed to help all learners achieve the lesson outcomes. Employ strategies such as Project-Based Learning to encourage participation.

1. **Step 1: Introduction to Autonomous Control and Efficiency**

   Begin by discussing the concept of autonomy in robots, that is, operating without constant human input. Link this to feedback control as the key enabler (Refer to Year 2 Manual, Section 8).

   Introduce the concept of efficiency in industry, providing simple local examples (e.g., sorting produce faster at a market, reducing waste in textile cutting, automating a step in shea butter production).

   Explain that the goal is to design a system that uses feedback to perform a task more efficiently.

To ensure differentiation and inclusivity:

a. Use simple analogies for feedback loops (e.g., maintaining a steady walking pace while carrying a load, adjusting water flow from a tap based on the level in a container).

b. Provide a list of potential local industry tasks where efficiency is important, perhaps categorised by perceived complexity, allowing learners to choose based on their readiness and interest.

c. Discuss different types of feedback control (e.g., simple on/off based on a sensor threshold versus more proportional responses, although complex control algorithms are not required at this stage) and how they might conceptually affect efficiency in different scenarios, linking to the Year 2 content.

d. Use visual examples of autonomous systems (e.g., self-driving vehicles, automated assembly lines, or even diagrams of home appliances with feedback mechanisms like a thermostat controlling a heater).

   e. Ensure explanations of "efficiency" and "autonomy" are clear and related to tangible, relatable examples from daily life or local contexts, using accessible language and avoiding overly abstract concepts initially.

2. **Step 2: Facilitating Problem Identification and Analysis**
   a. Guide groups through Learning Task 1 by helping them identify a specific, narrow task within a local industry that is suitable for improvement using a feedback system.
   b. Encourage them to move beyond simply stating a problem and to analyse the current inefficiencies in detail, that is, understanding the current process, identifying bottlenecks, and quantifying waste (time, material, energy) where possible.
   c. Provide resources where learners can find information about local industries (e.g., summaries of Ghana Statistical Service reports related to industry, Ghana business directories snippets focusing on small enterprises, articles or videos showcasing local manufacturing or agriculture).
   d. Consider inviting a guest speaker from a local industry or watching relevant short documentaries if feasible. If resources are available, learners can visit a relevant industry where feedback control is implemented to see firsthand how this is implemented.

To ensure differentiation and inclusivity:

   a. Provide descriptions of a few pre-selected local industry tasks with some initial data or observations on inefficiencies and ask groups to focus on deeper analysis and identifying the root causes of the inefficiencies within those specific contexts.
   b. Ask groups to attempt to quantify the inefficiency they observe or research (e.g., estimate the time taken per item, the percentage of wasted material, the number of errors per batch) to make their analysis more concrete and provide a clearer target for efficiency improvement.
   c. Provide support in understanding how to find or estimate such data.

3. **Step 3: Guiding Feedback System Design**

   This guides groups through Learning Task 2. This is a key design step. Help them identify precisely what variable(s) need to be sensed to understand the state of the task (the "controlled variable") and what actuator action is needed to control the task towards improved efficiency.

   Connect them back to the specific types of sensors and actuators they learned about in Year 1. Facilitate brainstorming of the simple feedback control logic – the rule(s) that link the sensor input to the actuator output to achieve the desired state or outcome efficiently.

   Encourage them to think about the desired state and how the robot would use feedback to maintain or reach that state.

To ensure differentiation and inclusivity:

a. Provide simplified scenarios with clear controlled variables, required sensor input types, and necessary actuator output types already identified.
b. Offer examples of how specific sensors are used in basic feedback loops through simple block diagrams or flowcharts, perhaps using examples from the Year 2 manual.
c. Ask learners to consider if their design needs to handle multiple inputs or outputs, or if the feedback loop requires slightly more complex conditional logic beyond a simple threshold.
d. Encourage them to sketch multiple design ideas and compare their potential effectiveness and feasibility in the local context before selecting one to develop further.

4. **Step 4: Supporting Documentation and Operational Sequence**

This guides learners through Learning Task 3. Provide a structured template for their design document, including clear headings for each required section (Problem Description, System Design, Feedback Mechanism, Operational Sequence, Justification of Efficiency Improvement).

Remind them of their skills in creating flowcharts and pseudocode from previous Year 3 lessons and require them to use one method consistently to represent the operational sequence of their feedback system.

Emphasise clearly explaining how the feedback loop, specifically, contributes to improving efficiency in their chosen local industry problem, providing a strong justification for their design choices by comparing the automated process to the current manual method.

To ensure differentiation and inclusivity:

a. Provide sentence starters or partial sentence frames for describing the problem, the proposed system, and explaining the feedback mechanism.
b. Offer examples of how to represent a simple feedback loop in both a flowchart and pseudocode format side-by-side, highlighting the key elements and transitions.
c. Encourage learners to include a detailed justification of their specific sensor and actuator choices based on the problem's requirements, the local context, and potential constraints.
d. Ask them to provide a more detailed explanation of the feedback logic, potentially including simple examples of how different sensor readings would trigger different actuator responses, demonstrating a deeper understanding of the control mechanism.

5. **Step 5: Presentation and Discussion of Prototype Designs**

a. Have groups present their identified local industry efficiency problem and their prototype feedback-driven autonomous system design.
b. Allocate sufficient time for each group to present their analysis of the problem and their proposed design.

c. Facilitate class discussion by asking clarifying questions about the identified inefficiencies, the functioning of the feedback mechanism, the choice of components, and how their design specifically improves efficiency compared to the current manual or less automated method.

d. Encourage learners to provide constructive feedback to their peers, focusing on the clarity of their problem analysis, the feasibility and effectiveness of their proposed feedback system design at a prototype level, and the strength of their justification for efficiency improvement.

e. Provide options for presentation format (e.g., verbal presentation with sketches on a board, a poster presentation, a digital slide presentation, or even a short video explaining their concept) to accommodate different learner strengths, learning styles, and available resources.

f. Ensure presentation space is accessible to all learners. Encourage respectful questioning and active listening from all learners throughout the presentations, fostering a collaborative learning environment.

KEY ASSESSMENT LEVEL 4: INDIVIDUAL PRACTICAL TASK

1. Instructions: Answer the following questions with clear explanations and justifications.

a. Consider a local cassava processing industry where peeling is done manually by workers. While experienced workers can achieve good results, the process becomes less efficient over time due to fatigue, variation in cassava size and shape, and the need to meet high production demand. These factors can lead to variations in peel thickness across batches, increased material loss when peeling at high speed and reduced productivity during long working hours.

i. Identify ONE specific variable that needs to be controlled for efficient peeling. Justify your choice. ***(2 marks***: 1 for variable, 1 for justification)

ii. For the variable identified in (a), propose the type of sensor required to measure it. Explain why this sensor is suitable. (***2 marks***: 1 for sensor type, 1 for explanation)

iii. Briefly describe the feedback control logic that would enable the system to autonomously adjust the peeling mechanism based on sensor input to reduce waste. (***3 marks***: for clear logic description)

b. By considering a block diagram of a generic feedback loop, which constitutes the following components: Desired State, Error, Processor/Controller, Actuator, System/Environment, Sensed State and Error, briefly explain how this loop facilitates autonomous operation for efficiency improvement in a real-world local industry scenario. Provide a specific example from Ghanaian industry, beyond the tomato sorting scenario, to illustrate your explanation. (***3 marks***: 2 for explaining autonomous operation via loop, 1 for a specific, relevant Ghanaian example).

2. **Individual Practical Task (*15 marks*)** - (Suggested Modes: Individual Project Assessment)

Project Title: Conceptual Design Documentation

Task: Document a conceptual prototype feedback-driven autonomous system for an identified local industry problem.

1. **Choose an Industry Problem:** Select a local industry problem where efficiency can be improved using a feedback-driven autonomous system. This can be the one from your group work or a different one.
2. **Document the Design:** Create a concise technical document (maximum 2 A4 pages) outlining your conceptual design. Your document must include the following sections with clear headings:
   a. **Problem Description (*3 marks*) :** Briefly describe the local industry, the specific inefficient task, and the nature of the inefficiencies (e.g., waste of time, material, energy, effort).
   b. Proposed Prototype System Concept (***4 marks***):
      i. Name your proposed system.
      ii. Provide a simple sketch or block diagram of the overall system.
      iii. List and briefly describe the proposed types of key components (sensors, processor, actuators) and their function. Justify your sensor and actuator choices based on the task and environment.
   c. Feedback Control Mechanism (***4 marks***):
      i. Draw a detailed block diagram of the primary feedback loop within your system.
      ii. Clearly label the sensed state, desired state (setpoint), error, processor/controller, and actuator action.
      iii. Describe the feedback control logic in clear, concise language (e.g., using an "if-then-else" structure or simple flow of thought).
   d. Autonomous Operation and Efficiency Improvement (***2 marks***):
      i. Explain how your designed feedback loop allows the robot to operate autonomously for the specific task without continuous human intervention.
      ii. Clearly articulate how this autonomous feedback control leads to specific efficiency improvements (e.g., increased speed, improved accuracy, reduced waste, enhanced safety) compared to the current method or a non-feedback approach.
   e. Operational Sequence Outline (***2 marks***):
      i. Provide either a flowchart or pseudocode illustrating the main steps of your system's autonomous operation, emphasising the feedback loop.

      (*15 marks*)

Conclusion

The week's lesson on designing feedback-driven autonomous systems was pivotal in applying prior knowledge to enhance operational efficiency in local Ghanaian industries. Key highlights include the conceptualisation of a prototype to address inefficiencies, such as manual tomato sorting, through a feedback loop involving sensors, actuators, and control logic. Learners analysed real-world inefficiencies, identified controlled variables, and designed systems to improve speed, consistency, and resource use. By outlining operational sequences via flowcharts or pseudocode, the lesson emphasised practical automation solutions, fostering skills to create impactful, autonomous systems tailored to local industrial needs.

PLC Session 4

45 minutes	Planning & Reviewing the Learning Plan

1. Begin the session by recapping lessons from the previous year/session/week, focusing on how learning was applied.
2. Use your subject specific app to explore the content before planning. Ask the app to
   a. explain the week's concepts in a practical way that will make it easier for learners to understand
   b. identify two or three common misconceptions learners may have on the week's content and how to address them using contexts familiar to Ghanaian learners. (NTS 3a, 3g and 3m).
3. Use your subject specific app to plan your lesson. Ask the app to
   a. suggest the most effective method for teaching the week's concepts to learners with diverse backgrounds.
   b. suggest learning activities, differentiation and essential questions based on the method of teaching (NTS 3a-3q)
   c. review your prepared learning plan and address any challenging issues you may have identified in your subject area (NTS 3a-3c).

Note

1. Take note of the recap topics linked to Years 1 and 2 while preparing for your lesson.
2. Refer to the Values Learning Community Handbook and link the values for the week in your lesson to ensure that learners are applying what they learn.
3. Plan the week's key assessment with your app

a. The key assessment for the week is ***individual Practical task***. Chat with your app to plan this assessment, indicating what you and your learners will do before, during and after the assessment (NTS 3k-3q).

10 minutes	Content Exploration

4. As a subject group, select one problem or idea/term in the week's content that is challenging (not a routine procedural question, but one that requires conceptual understanding) and then
   a. use the app, ask it to solve the problem step by step and explain the reasoning behind each step (you may be assisted by a teacher who is strong in the content area)
   b. all teachers in the group work through a similar problem individually for 3–4 minutes
   c. the group then works through the problem together step by step
   d. highlight the most common errors made at each step and discuss (NTS 1c, 1e, 2c, 2e, 3e-3j).

Note

1. This part is not just about 'How do I teach this?' but how do I teach it for learners to understand?
2. If the group still shares the same content gap, use the app: ask it to solve the problem step by step and explain the reasoning behind each step.
3. Problems/ideas discussed should be sourced from the Teacher Manual content, not invented separately. This keeps the content exploration directly connected to what teachers will deliver.

30 minutes	Enactment

5. Enact a teaching activity based on your learning plan (NTS la-1b, 1e-lg, 2c, 2e, 2f, 3c-3j and 3o).
   a. Colleagues observe with specific attention to clarity of concept explanation, use of familiar Ghanaian examples and how misconceptions are handled if they arise.
   b. After the enactment, ask for structured feedback from the group
      i. was the concept explained in a way that a learner with no prior knowledge could follow?
      ii. were the examples grounded in contexts familiar to Ghanaian learners?
      iii. did any misconceptions surface and if so, how were they addressed?

5 minutes	Reflection

6. Reflect and share your views on the session. One thing that worked well, what did not work well and what you will do differently. (NTS 1a, 1b).

7. Identify a critical friend to observe and provide feedback on your lesson in relation to the PLC Session conducted (NTS 1f, 1g).
8. Remember to
   a. transfer your chats into your learning planner template.
   b. read your Teacher Manual and related Learner Material in preparation for the next session. Take note of the recap weeks, which are linked to Years 1 and 2 (NTS 3a).
   c. integrate National Values, GESI, SEL, and 21st century skills into your learning activities, assessment tasks, and rubrics/mark schemes.

WEEKS 5 & 6

Learning Indicator: *Observe and reverse-engineer the controls of a functional autonomous robotic system*

FOCAL AREA: REVERSE ENGINEERING AUTONOMOUS ROBOT CONTROLS

Introduction

This lesson advances understanding of robotic systems by examining their internal workings. Attention is directed towards dissecting the control mechanisms of existing functional autonomous robotic systems and robots. Learners will focus on careful observation and methodical analysis to deduce how these systems sense their environment, process information, and execute actions independently. This process of reverse engineering will deepen comprehension of autonomous behaviour and the intricate design of control systems within real-world robotic applications.

Building on Foundational Concepts

The reverse engineering knowledge content systematically builds upon prior foundational concepts. Learners apply their understanding of robot subsystems from Year 1, Week 3, to identify actual sensors, processors, and actuators within functional systems. Concepts of feedback loops from Year 1, Week 4, and Year 2, Week 3, become essential for deducing how autonomous control is achieved. Furthermore, skills in outlining control logic through algorithms, pseudocodes, and flowcharts from Year 1, Week 17, and Year 2, Week 17, are now employed to infer and represent the internal decision-making processes of observed robots. This lesson thus consolidates theoretical knowledge with practical analysis.

Recap Weeks

Year 1	**Week 3** 1. Describe the attributes and functionalities of a robot's subsystems and how they interconnect **Week 4** 1. Contrast non-feedback loop systems and feedback loop systems. **Week 7** 1. Discuss the scientific principles underlying the operation of sensors. **Week 17** 1. Determine the Inputs, Processes and Outputs required to solve a particular problem. 2. Define solutions to basic automated and robotic problems using algorithms, pseudocodes, and flowchart diagrams. **Weeks 19 & 20** 1. Create robots using robotic kits and/or local materials to implement basic mechanics for actuations…" (specifically "ARMS, WINGS & OTHERS" – Gripping Figures, Lifting Mechanisms)

Year 2	**Week 3** 1. Utilise component and system diagrams to design a feedback system for implementing controls in given scenarios. **Week 17** 1. Define solutions for control and feedback systems using algorithms, pseudocodes, and flowchart diagrams. **Weeks 19 & 20** 1. Create robots using robotic kits and/or local materials to implement basic mechanics for actuations that make use of…" (specifically "ARMS, WINGS & OTHERS" – Shooting robots, Automatic Doors, Raking up or out, Creating Wind)

KNOWLEDGE CONTENT

In order that the learning and further still great innovations will be achieved, sight must not be lost on the foundation which is the relevant topics in mathematics physics; such as geometry, calculus, equation of a straight line for calibration of the the light sensor for example, reflectivity of light, absorption of light, colours and the wavelengths, ultrasonic waves, etc. For Reverse engineering or for that matter any 'strand' of engineering to be understood and be most impactful the relevant mathematics and physics topics ought to be well within the grasp of the learner

Understanding Reverse Engineering in Robotics

Reverse engineering, in the context of robotics, is the process of disassembling or analysing a functional system to understand its components, internal mechanisms, and, crucially, its control logic and algorithms. It is not simply about taking things apart, but rather about inferring the design choices, operational principles, and decision-making processes that enable the robot to perform its autonomous functions. For autonomous systems, this involves deducing how the robot achieves independence from continuous human guidance.

Thus, the purpose of reverse engineering controls can be captioned under the following:

1. **Deepen Understanding:** To gain profound insight into how theoretical control principles (like feedback loops) are applied in practical, working systems.
2. **Learn from Existing Designs:** To identify effective design patterns, component choices, and control strategies used by engineers.
3. **Problem Identification:** To understand why a system might behave in certain ways, identify limitations, or pinpoint areas for potential improvement.
4. **Inspiration for New Designs:** The insights gained can inform the development of novel robotic solutions for local challenges.

Characteristics of a Functional Autonomous Robotic System

A functional autonomous robotic system possesses specific characteristics that enable its independent operation. When reverse engineering, these are the behaviours and underlying mechanisms to look for:

1. **Perception (Sensing):** The ability to gather information about its internal state and external environment. This relies on sensors.
2. **Cognition (Processing):** The ability to interpret sensed data, make decisions, plan actions, and execute control logic. This is performed by a processor or controller.
3. **Action (Actuation):** The ability to interact physically with its environment based on decisions made. This relies on actuators and mechanical structures.
4. **Adaptability/Responsiveness:** The capacity to adjust its behaviour in response to changes or unexpected events in its environment, often through feedback.
5. **Goal-Oriented Behaviour:** The system operates to achieve a defined objective without constant human instruction.

The Reverse Engineering Process for Controls

The reverse engineering of a robot's controls is a systematic process. It moves from observing external behaviour to inferring internal logic.

1. **External Observation and Behavioural Analysis:** First carry out an external evaluation of the behaviour of the system or robot by following these steps
   - **a.** Begin by observing the robot or the robotic system's overall operation. What task does it perform? How does it interact with its surroundings?
   - **b.** Identify its inputs (what it senses from the environment) and its outputs (what actions it takes).
   - **c.** Note how its behaviour changes under different environmental conditions or inputs. This provides initial clues about its decision-making.

For example, observing an automated hand-washing station at a market. Observe to find answers to questions such as: How does it sense hands? When does it dispense soap/water? When does it stop? Does it adapt if hands are not fully inserted?

2. **System Decomposition and Component Identification:** Identify the major physical components and subsystems. Which parts look like sensors? Which are actuators? Where is the central processing unit likely located?

Hint

- *Refer to Year 1, Week 3 (Learning Indicator 1.2.1.2) on the attributes and functionalities of robot subsystems and their interconnection.*

Note

- Not all components may be immediately obvious or accessible, but categorising them by function (sensing, processing, actuating) is key.

3. **Data Flow and Feedback Loop Tracing:** Analyse how information is likely transmitted from sensors to the processor, and from the processor to actuators. Also, identify potential feedback loops by asking the question, "What sensor provides information that directly influences an actuator's action to maintain a desired state?"

Recall

The concepts of feedback versus non-feedback systems from Year 1, Week 4 (Learning Indicator 1.2.2.1) and designing feedback systems using component diagrams from Year 2, Week 3 (Learning Indicator 1.2.1.2) are crucial here. These skills are now applied in reverse.

4. **Inferring Control Logic, Circuits, and Algorithms:** This is the most challenging and insightful step. Beyond high-level software programming logic (using if-then-else conditionals and loops to map sensor inputs to actuator commands), reverse engineering must extend to the underlying digital circuitry. Instruct learners to observe physical stimulus-response patterns to systematically map input-output combinations into a system truth table. From this table, learners should deduce the basic combinational logic circuits (utilising AND, OR, and NOT gates) that hardwire the robot's fundamental safety, override, or control behaviours before software instructions are executed.
5. **Conceptual Reconstruction and Documentation:** Based on the analysis, create diagrams (block diagrams, flowcharts) and pseudocode to represent the deduced control system. This is a conceptual "rebuild" of how the system operates. Clearly document the identified components, their roles, the traced data flow, and the inferred control logic. This serves as the output of the reverse engineering process.

Tools and Techniques for Reverse Engineering Controls

While actual disassembly might not always be feasible or safe, conceptual reverse engineering heavily relies on:

1. **Systematic Observation:** Paying close attention to how the robot behaves under various conditions.
2. **Diagramming:** Using block diagrams to represent subsystems and their interconnections, and flowcharts to represent the control logic.
3. **Pseudocoding:** Translating observed behaviours and inferred rules into structured pseudocode.
4. **Hypothesis Testing:** Forming hypotheses about how a system works and then testing them through observation or by simulating conditions, if possible.

Things to Note

Always remember safety first! If working with physical systems, ensure all power is disconnected and proper safety protocols are followed.

Learning Tasks

These tasks guide learners through the process of observing and reverse-engineering the controls of a functional autonomous robotic system. Depending on the available time or resources, administer one or more of the following learning tasks to help learners reinforce understanding and acquire new knowledge or skills.

1. Watch a functional autonomous robotic system or robot (e.g., an automated hand-washing station, a simple line-following robot, a light-sensitive automatic gate, a simple sorting machine).
    a. Pay close attention to its overall operation, what triggers its actions, and how it reacts to different environmental conditions.
    b. List the inputs the system receives from its environment (what it senses) and the outputs it produces (what actions it performs).
    c. From observation, identify and list the likely sensor(s) and actuator(s) present, and briefly describe their probable functions in the system.
2. For the observed system:
    a. Deduce the primary feedback control loop that enables its autonomous function.
    b. Sketch a simple block diagram of this deduced feedback loop. Clearly label
        i. The specific physical state or variable being sensed.
        ii. The type of sensor(s) used.
        iii. The desired state or set point.
        iv. The action the actuator(s) perform.
        v. How the processor/controller likely uses the sensor input and desired state to determine the actuator output.
    c. Briefly explain how this specific feedback loop allows the system to operate autonomously and adapt to its environment.
3. Based on your observations and the identified feedback loop:
    a. Infer the core control logic governing the system's autonomous behaviour.
    b. Express this deduced control logic using either a flowchart or pseudocode. Ensure it clearly shows decision points and sequences of actions based on sensed information.
4. Document your reverse engineering findings in a structured report. It should include the following:
    a. A description of the observed autonomous system and its function.

b. A list of identified components (sensors, actuators, estimated processor type) and their roles.
c. The block diagram of the primary feedback loop with clear labels.
d. A narrative explanation of the feedback loop's operation.
e. The flowchart or pseudocode represents the inferred control logic.
f. A concluding statement on what was learned about autonomous control from reverse engineering this system.

PEDAGOGICAL EXEMPLARS

This section guides the teacher in facilitating the process of reverse engineering robotic controls, focusing on observation, deduction, and documentation. Differentiated approaches are suggested to cater to diverse learning needs and ensure inclusivity.

1. **Step 1: System Observation and Initial Identification**
   a. **Objective:** To introduce reverse engineering through direct observation and guide learners in identifying basic system inputs, outputs, sensors, and actuators (Learning Task 1).
   b. **Activity:** Begin by presenting a functional autonomous robotic system. This can be a physical device (e.g., an automatic hand-washing station, a simple toy robot that detects edges), a detailed video demonstration of a local automated process (e.g., a simulated gate opener that reacts to vehicle presence, a simple sorting machine for local produce), or an interactive simulation.
   c. **Guidance:** Instruct learners to observe the system's behaviour carefully, noting how it responds to environmental changes. Prompt them to identify what the system "sees" or "feels" (inputs) and what it "does" (outputs). Guide them in identifying the physical components responsible for sensing and acting.

To ensure differentiation and inclusivity:

a. For learners requiring additional support, provide a checklist of common robotic inputs (light, sound, touch, distance) and outputs (movement, sound, light, dispense).
b. For visual learners, ensure clear visibility of the system's operation; for auditory learners, describe actions and sounds.
c. Encourage verbal description of observations for learners who may find written listing challenging.

2. **Step 2: Deconstructing the Feedback Loop**
   a. **Objective:** To guide learners in deducing the primary feedback control loop and sketching its block diagram (Learning Task 2).
   b. **Guidance:** Direct learners to focus on the identified inputs and outputs, then trace how the system's actions are influenced by its perceptions. Facilitate discussion on

what the system is trying to maintain or achieve (the desired state) and how it uses sensor information to adjust. Guide them to sketch a simple block diagram of this loop, ensuring all key parts: sensed state, desired state, sensor, processor/controller, and actuator, are clearly labelled

To ensure differentiation and inclusivity:

i. Provide a blank block diagram template for learners to fill in, aiding visual learners and those needing structure.

ii. **Ask guiding questions:** "What is the robot trying to keep constant?" (desired state), "How does it know its current condition?" (sensed state), "What is the action it takes to correct itself?" (actuator).

3. **Step 3: Inferring Control Logic**

a. **Objective:** To support learners in inferring the core control logic and expressing it using a flowchart or pseudocode (Learning Task 3).

b. **Guidance:** Based on their observations and the deduced feedback loop, prompt learners to think about the "rules" the system follows. Encourage them to articulate the decision-making process: "If [sensor condition] is met, then [actuator action] occurs." Guide them in translating these inferred rules into structured pseudocode or a flowchart.

To ensure differentiation and inclusivity:

i. Offer simple "if-then-else" pseudocode structures or basic flowchart symbols as starting points.

ii. For learners who thrive on collaboration, encourage small group discussions to brainstorm the logic before individual representation.

4. **Step 4: Documenting Findings**

a. **Objective:** To facilitate the systematic documentation of reverse engineering findings into a structured report (Learning Task 4).

b. **Guidance:** Provide a clear outline or template for the report, specifying all required sections (description of system, identified components, feedback loop diagram, explanation of loop operation, control logic representation). Emphasise clarity, precision, and logical flow in their explanations.

To ensure differentiation and inclusivity:

i. Provide sentence starters or partial sentence frames for each section to support learners in their writing.

ii. Offer a checklist for report completion, ensuring all necessary elements are included.

iii. Encourage peer review within groups to provide constructive feedback on clarity and completeness before final submission.

iv. For learners with fine motor challenges, allow typed reports or provide assistance with diagram drawing tools. Ensure templates are accessible.

KEY ASSESSMENT: MID-SEMESTER EXAMINATION

1. **Instructions:** Answer the following questions with clear explanations and justifications.
   - **a.** Consider a functional autonomous robotic system designed to open and close a chicken coop door based on ambient light levels (opens at dawn, closes at dusk) to protect chickens from predators.
     - **i.** Identify the primary sensor and actuator in this system. Justify your choices. (*2 marks*)
     - **ii.** Describe the core feedback loop of this system, indicating the sensed state, desired state, and the general action taken by the actuator. (*3 marks*)
     - **iii.** Using pseudocode or a flowchart, outline the main control logic you would infer for this autonomous chicken coop door. (*3 marks*)
   - **b.** During the reverse engineering of an autonomous street light, you observe that the light turns on when it gets dark and turns off when it gets bright. You identify a light-dependent resistor (LDR) and a light bulb. Explain how the process of deducing the control logic for this system relies on your prior knowledge of feedback systems and basic programming concepts. (*2 marks*)

2. **Group-Based Task/Project (*25 marks*)**

Project Title: Conceptual Reverse Engineering Project

Task: As a group, observe a given functional autonomous robotic system (or a detailed video demonstration/simulation), reverse-engineer its control mechanisms, and present your findings.

1. Carefully observe the provided functional autonomous robotic system (e.g., a simple automatic soap dispenser, an obstacle-avoiding robot, a simulated traffic light system). Pay attention to its behaviour under different conditions.
2. Identify and list the main components (sensors, actuators, and processor) and describe their probable functions in the system.
3. Deduce the primary feedback loop(s) of the system. Draw a clear block diagram of this loop, labelling all parts (sensed state, desired state, processor, actuator, system/environment, error).
4. Infer the core control logic that governs the system's autonomous operation. Represent this logic using either a detailed flowchart or pseudocode.
5. Prepare a presentation (10-15 minutes) to explain your reverse engineering findings to the class. The presentation should:
   - **i.** Introduce the observed system and its autonomous function.
   - **ii.** Show your identified components and their roles.
   - **iii.** Explain the deduced feedback loop using your diagram.
   - **iv.** Present and explain your inferred control logic (flowchart/pseudocode).

v. Discuss what your group learned about autonomous control design from this exercise. **(*25 marks*)**

Hint

- *Conduct **Mid-semester Examination**. It should cover contents taught from Week 1 to 5. Refer to the **Table of Specification and Structure of the Paper**. Learners' papers should be marked immediately and scores entered into the SAP.*

Structure

Section A: 25 Multiple-choice questions

Section B: 2 Essay Questions

Table of specification for item construction for section A and B

Week	Learning Indicator	Types of questions	DoK Levels				Total
			1	2	3	4	
1	Identify both local and global problems that can be fixed using robots and automated systems	MCQs	2	2	-		4
		ESSAY	-	-	-		-
2	Identify, document and present on problems in society and local industries and suggest the role robots can play in solving these problems	MCQs	2	2	2		6
		ESSAY	-	-	1		1
3	Prepare technical reports including flowcharts or pseudo codes that outline sequence of steps to solve problems Peer-review and critique the technical reports	MCQs	-	2	1		3
		ESSAY	-	-	-		
4	Design a prototype feedback-driven autonomous system for efficiency improvement in an identified local industry	MCQs	3	2	1		6
		ESSAY					
5 & 6	Observe and reverse-engineer the controls of a functional autonomous robotic system	MCQs	1	3	2		6
		ESSAY			1		1
		Total	8	11	8		27

Conclusion

The lesson on reverse engineering autonomous robotic controls was well crafted to deepen learners' grasp of system functionality. Through systematic observation, the intricate behaviours of robotic systems were analysed, revealing how sensors, processors, and actuators interplay to enable autonomy. Key highlights include deducing feedback loops, as exemplified in systems like automated hand-washing stations, and inferring control logic through flowcharts or pseudocode. By documenting findings, learners gained insight into design principles, enhancing their ability to innovate solutions for local challenges, thus reinforcing the practical application of autonomous control systems in real-world contexts.

PLC Session 5 & 6

45 minutes	Planning & Reviewing the Learning Plan

1. Begin the session by recapping lessons from the previous session, focusing on how learning was applied.
2. Use your subject specific app to explore the content before planning. Ask the app to
   a. explain the week's concepts in a practical way that will make it easier for learners to understand.
   b. identify three common misconceptions learners may have on the week's content and how to address them using contexts familiar to Ghanaian learners (NTS 3a, 3g and 3m).
3. Use your subject specific app to plan your lesson. Ask the app to
   a. suggest the most effective method for teaching the week's concepts to learners with diverse backgrounds.
   b. suggest learning activities, differentiation and essential questions based on the method of teaching (NTS 3a-3q).
   c. review your prepared learning plan and address any challenging issues you may have identified in your subject area (NTS 3a-3c).

Note

- Take note of the recap topics linked to Years 1 and 2 while preparing for your lesson.
- Refer to the Values Learning Community Handbook and link the values for the week in your lesson to ensure that learners are applying what they learn.

4. Use your app to plan and administer the mode of assessment for the Student Assessment Portal (SAP).
   a. The key assessment for the week is ***mid-semester examination***. Chat with your app to plan this assessment by reviewing or developing the assessment items/tasks and marking schemes/rubrics for the mid-semester examination (NTS 3k-3q).

10 minutes	Content Exploration

5. As a subject group, select one problem or topic/idea/term in the week's content that is challenging (not a routine procedural question, but one that requires conceptual understanding) and then

a. use the app, ask it to solve the problem step by step and explain the reasoning behind each step (you may be assisted by a teacher who is strong in the content area)

b. all teachers in the group work through the problem individually for 3–4 minutes

c. the group then works through it together the problem, step by step

d. highlight the most common errors made at each step (NTS 1c, 1e, 2c, 2e, 3e-3j).

Note

- This part is not 'How do I teach this?' but how do I teach it for learners to understand?
- If the group still shares the same content gap, use the app: ask it to solve the problem step by step and explain the reasoning behind each step.
- Problems/ideas discussed should be sourced from the Teacher Manual content, not invented separately. This keeps the content exploration directly connected to what teachers will deliver.

30 minutes	Enactment

6. Enact a teaching activity based on your learning plan (NTS la-1b, 1e-lg, 2c, 2e, 2f, 3c-3j and 3o).

   a. Colleagues observe with specific attention to clarity of concept explanation, use of familiar Ghanaian examples and how misconceptions are handled if they arise.

   b. After the enactment, ask for structured feedback from the group

      i. was the concept explained in a way that a learner with no prior knowledge could follow?

      ii. were the examples grounded in contexts familiar to Ghanaian learners?

      iii. did any misconceptions surface and if so, how were they addressed?

5 minutes	Reflection

7. Reflect and share your views on the session. One thing that worked well, what did not work well and what you will do differently. (NTS 1a, 1b).

8. Identify a critical friend to observe and provide feedback on your lesson in relation to the PLC Session conducted (NTS 1f, 1g).

9. Remember to

   a. transfer your chats into your learning planner template.

   b. read your Teacher Manual and related Learner Material in preparation for the next session. Take note of the recap weeks, which are linked to Years 1 and 2 (NTS 3a).

   c. integrate National Values, GESI, SEL, and 21st century skills into your learning activities, assessment tasks, and rubrics/mark schemes.

WEEKS 7 & 8

Learning Indicators

1. *Test operational robots for specified tasks, observe/document identified flaws*
2. *Identify faults in robotic systems and trace them to algorithm flaws, design flaws or coding errors*
3. *Go through series of iterations in fixing the identified flaws or improve robot's performance*

FOCAL AREA: ROBOT TROUBLESHOOTING – DIAGNOSIS TO IMPROVEMENT

Introduction

Robots, being complex systems, can encounter issues. This lesson focuses on developing a structured approach to identify, diagnose, and resolve these issues, known as troubleshooting. Learners will apply a systematic routine to test operational robots, categorise identified flaws, and implement iterative corrections, enhancing their skills in ensuring robot reliability and performance.

Building on Foundational Concepts

This section significantly expands upon Year 2's troubleshooting by applying its systematic routine to real-world robot testing. It builds on Year 2, Week 23's "Be Observant" by detailing methods like sensor data monitoring (rooted in Year 1, Week 7 sensors) and actuator response monitoring (from Year 1 and 2 actuators). Furthermore, it deepens the Year 2, Week 23 concept of isolating problems by thoroughly categorising flaws into algorithm, design, or coding errors, which are also types of issues introduced in Year 2. Finally, the iterative "test-fix-retest" cycle is a direct progression from Year 2, Week 24's "Fixing Identified Flaws", emphasising structured documentation crucial for learning and future reference.

Recap Weeks

Year 1	**Week 3** 1. Describe the attributes and functionalities of a robot's subsystems and how they interconnect **Week 7** 1. Identify different types of sensors and actuators **Week 17** 1. Determine the Inputs, Processes and Outputs required to solve a particular problem. 2. Define solutions to basic automated and robotic problems using algorithms, pseudocodes, and flowchart diagrams. **Weeks 19 & 20** 1. Create robots using robotic kits and/or local materials to implement basic mechanics for actuations (e.g., Gripping Figures, Lifting Mechanisms).

Year 2	Week 23 & 24
	1. Observe and identify flaws in an operational robot and trace them to either algorithm flaws, design flaws or coding errors. 2. Use a series of iterations to fix the identified flaws or improve the performance of solutions.

KNOWLEDGE CONTENT

Testing Operational Robots and Identifying Flaws

Before a robot can be reliably deployed, especially in contexts such as local industries or community services, it must undergo rigorous testing. Testing is a systematic and disciplined process to evaluate a robot's performance against specified requirements and to uncover any deviations or shortcomings. It is a critical initial phase of the troubleshooting routine.

The following are some reasons for carrying out rigorous testing on robots and robotic systems:

1. **Reliability:** Ensures the robot consistently performs its task as expected, preventing unexpected failures or downtime in critical applications like autonomous refuse collection or automated farming.
2. **Performance:** Verifies that the robot meets predefined efficiency metrics, such as the speed at which a delivery drone transports goods in a Ghanaian market, the accuracy of a sorting machine for cocoa beans, or the throughput of an assembly robot.
3. **Safety:** Identifies behaviours that could pose risks to people, property, or the environment. For instance, testing ensures that an autonomous vehicle correctly detects and avoids pedestrians or that a robotic arm operates within safe limits.
4. **Validation:** Confirms that the robot's design and programming genuinely meet the original problem's requirements and user needs. This ensures the engineered solution effectively addresses the initial challenge.
5. **Optimisation:** Provides empirical data that can be used for continuous improvement, leading to enhanced efficiency (e.g., reduced power consumption), better resource utilisation, or increased robustness in varying environmental conditions.

Designing a Comprehensive Test Plan

A structured test plan is not merely a checklist; it is a strategic document that guides the entire evaluation process. It ensures that testing is thorough, repeatable, and yields actionable data, preventing the reliance on mere casual observations. A comprehensive test plan would usually contain the following:

1. **Test Objectives:** These clearly define what specific aspects of the robot's performance are being evaluated. For example, an objective might be: "To verify the accuracy of the robot's object recognition system under varying lighting conditions" or "To assess the robot's ability to navigate a defined obstacle course within a specified time limit."
2. **Test Environment:** A detailed description of the conditions under which the test will be conducted is vital. This includes physical parameters (e.g., lighting levels, surface type,

presence of specific obstacles, temperature) and any relevant external factors. For instance, testing a farming robot might require setting up scenarios with uneven terrain, simulated rain, or varying plant densities.

3. **Test Cases/Steps:** These are precise, repeatable instructions for operating the robot for the specific task. Each test case should represent a specific scenario or input condition that the robot might encounter. For a line-following robot, test cases might include "following a straight line," "navigating a 90-degree turn," or "crossing an intersection." These steps must be clear enough for any individual to replicate precisely.
4. **Expected Outcomes:** For each test step, the anticipated behaviour or result if the robot performs correctly must be clearly defined. This provides a baseline against which actual performance can be compared. Expected outcomes should be quantifiable where possible (e.g., "Robot completes the maze in less than 3 minutes," "Robot sorts 95% of red items correctly," "Robot stops within 5 cm of the target").
5. **Observation and Documentation Methods:** This specifies precisely how performance will be measured and recorded during testing. This includes objective metrics (e.g., task completion time, percentage of successful sorts, number of collisions) and a structured format for noting observations and identified flaws.

Effective Methods of Observation and Documentation

During testing, close observation and accurate documentation are paramount. A "flaw" or "shortcoming" is any deviation from the expected or desired performance. These methods align with the "Be Observant" step from Year 2, Week 23.

Recall

Remember that while dealing with troubleshooting in Year 2, Weeks 23 & 24, it was stated that "Observation is key to effective troubleshooting."

Some effective methods of observation used in testing or troubleshooting a robot include:

1. **Visual Inspection:** This involves direct, careful observation of the robot's physical state and movements. One observes the robot's overall behaviour, the smoothness or jerkiness of its motion, and whether components appear damaged, misaligned, or obstructed. For example, when observing an automated refuse collection robot, one would visually check if its wheels were turning freely, if its grabbing arm is reaching correctly, or if any debris is visually obstructing its path. This method is crucial for detecting obvious physical issues and understanding general operational patterns. It directly relates to the "Sight" aspect of observation, emphasised in Year 2, Week 23.
2. **Sensor Data Monitoring:** This method focuses on examining the raw data that sensors are feeding to the robot's controller. This might involve using a debugger, a serial monitor, or a dedicated software interface to view real-time values from light sensors, distance sensors, or encoders. By observing sensor data, one can determine if the robot is accurately perceiving its environment or if a sensor is faulty, providing erroneous readings, or perhaps requires re-calibration. For instance, if an autonomous irrigation system is not

watering plants, monitoring its soil moisture sensor data can reveal if it is incorrectly reading the soil as moist when it is actually dry. This reinforces principles from Year 1, Week 7 on sensor operation.

3. **Actuator Response Monitoring:** This entails observing how the robot's actuators (motors, servo motors, grippers, pumps) respond to commands from the controller. This can involve watching for specific movements, listening for unusual sounds (e.g., grinding or clicking from motors as mentioned in Year 2, Week 23), or feeling for unexpected vibrations or heat. For example, if a robot arm designed to pick up items fails to do so, monitoring its gripper actuator can confirm if the command is being sent, if the motor is attempting to close, or if it lacks sufficient power. This connects directly to understanding the function of actuators as covered in Years 1 and 2.
4. **Timing:** This involves precisely measuring the duration of specific actions, subtasks, or the overall completion time of a given task. This is critical for evaluating the robot's efficiency and responsiveness. For example, if a robot designed to sort produce takes an unexpectedly long time to complete a batch, timing each sorting cycle can pinpoint which specific step (e.g., scanning, moving, releasing) is causing the delay. Discrepancies in timing can indicate bottlenecks, processing delays, or mechanical inefficiencies that require further investigation.
5. **Error Counting:** This method involves quantifying the number of times a specific error or undesired behaviour occurs over a series of test runs. By systematically counting occurrences of a particular flaw (e.g., how many times a line-following robot drifts off the line, how many items a sorting robot misplaces, or how many times a delivery robot collides with an obstacle), patterns can be identified. This provides objective data to measure the severity and frequency of a flaw and track whether iterative fixes are truly improving the robot's reliability.

Tracing Shortcomings to Root Causes

Once flaws are carefully documented, the next crucial step is to diagnose their root cause. This process involves applying the "Isolate the Problem" and "Understand the System" steps from **Year 2, Week 23**. Refer to these sections for the detailed steps

Recall

From Year 2, Weeks 23 & 24 - "There are typically three broad categories of issues in robotics: algorithm flaws, design flaws, and coding errors. Each type of problem requires a different troubleshooting approach"

As highlighted in Year 2, Week 23, it is sometimes easy to confuse algorithm flaws with coding errors. An algorithm flaw impacts the overall approach and logic used to perform a task; even perfectly bug-free code will still execute a flawed strategy. A coding error is a mistake in how that algorithm is translated into a working program; it involves correcting issues at the syntax or command level. An algorithm flaw is a conceptual problem; a coding error is an implementation problem.

Iterative Fault Correction and Performance Improvement

Robotics development is rarely a linear process. It is an iterative cycle of continuous refinement, where solutions are progressively implemented and validated. The "test-fix-retest" cycle is fundamental to achieving robust and efficient performance, acknowledging that fixing one issue might reveal another or open opportunities for optimisation. This directly builds on **Year 2, Week 24 ("Fixing Identified Flaws")**.

The Iterative Cycle for Robot Improvement is as follows:

1. **Test:** Execute the designed test plan, observe the robot's behaviour thoroughly, and meticulously document any identified flaws or deviations from expected performance. This is the starting point for gathering data.
2. **Identify:** Clearly pinpoint the specific problematic behaviours observed during testing. This moves beyond general observations to precise descriptions of what is going wrong.
3. **Trace:** Systematically diagnose the root cause of each identified flaw, categorising it as an algorithm, design, or coding error. This step requires applying the troubleshooting steps from **Year 2, Week 23** (Isolate the Problem, Understand the System).
4. **Fix:** Implement a specific change aimed at addressing the identified root cause. This could involve modifying the algorithm (e.g., adjusting control parameters), altering a physical design aspect (e.g., reinforcing a joint), or correcting a line of code (e.g., fixing a typo or logic error).
5. **Re-Test:** After implementing a fix, it is imperative to run the relevant tests again. This verifies whether the original flaw has been resolved, and equally important, checks if any new issues have been inadvertently introduced (known as "regression testing").
6. **Document:** Throughout the entire cycle, every step must be thoroughly documented. This includes recording the flaw addressed, the exact change implemented, the detailed results of the re-test, and any new issues that arose. This forms a crucial part of the learning process and is a valuable resource for future reference.
7. **Repeat:** The process is repeated until the robot performs as required, achieves optimal performance, or until new flaws are no longer being introduced by previous fixes. This highlights that robotics development is a continuous journey of refinement.

Documentation of Iterations for Structured Reporting

Maintaining a detailed log of all changes and their outcomes is critical for effective troubleshooting and for learning from the process. This directly refers to the Technical Document Structure for Troubleshooting and Error Correction introduced in Year 2, Weeks 23 and 24.

For each iteration, the following information should be recorded, which includes:

1. The specific flaw being addressed.
2. The exact change implemented (e.g., "Changed line 45 of the text-based code to speed = 100 from speed = 50," or "Added a counterweight to the robot's base," or "Introduced a PID controller to the line-following algorithm").

3. The detailed results of re-testing (e.g., "Robot now moves faster, but sometimes overshoots target," or "Drifting on turns is reduced by 70%," or "Collision avoided consistently").
4. Any new issues that have arisen as a result of the fix?
5. This documentation forms key sections of the comprehensive technical document, particularly under "Iteration Process for Fixing" and "Testing and Validation."

Crucial Considerations During Iteration

The following are critical considerations to consider when carrying out iterative tests:

1. **One Change at a Time:** To isolate the effect of a modification, it is best practice to implement only one significant fix or change at a time. This allows the troubleshooter to clearly identify whether the change resolved the intended flaw or introduced new ones.
2. **Regression Testing:** Always ensure that fixing one flaw does not inadvertently introduce new issues in previously working functions. A thorough re-test should cover not just the problematic area but also other critical functionalities.
3. **Performance Improvement vs. Error Correction:** The iterative process is not solely about fixing errors. It is also an opportunity for continuous performance improvement, optimising the robot's efficiency, speed, accuracy, or robustness even if no explicit "flaw" is present. This aligns with the overall "evaluating and improving" aspect of the content standard.

Learning Tasks

These tasks will guide you through the practical application of **robot troubleshooting**, building on the systematic approach and technical documentation skills you learned in Year 2. You will engage in evaluating robot performance, identifying specific flaws, determining their root causes, and iteratively implementing solutions to improve functionality

1. This task focuses on applying systematic testing methods to observe and record any operational issues a robot may exhibit.
   a. **Robot and Task Orientation:** You will be provided with an operational robotic system and a specific task it is designed to perform. Carefully observe the robot to understand its intended functions, expected movements, and the overall goal of its assigned task.
2. **Test Plan Development:** Design a comprehensive test plan for the robot's specified task. Your plan must include:
   a. Clearly stated test objectives that define what aspects of the robot's performance are being evaluated.
   b. Detailed, step-by-step test cases that outline specific actions or scenarios the robot will undergo.
3. The expected outcome for each test case if the robot performs correctly.

4. A method for observing and documenting the robot's behaviour, including any deviations from the expected outcomes.

   a. **Test Execution and Flaw Recording:** Execute your designed test plan meticulously. As the robot performs its task, observe its behaviour closely. Whenever the robot deviates from the expected outcome or exhibits any unexpected behaviour, document it as a flaw. For each identified flaw, record:

      i. A precise description of what happened (the observed behaviour).

      ii. When it happened (the specific point in the task or the environmental conditions present).

      iii. How frequently it occurred (if applicable, noting if it's intermittent or consistent).

      iv. Any other relevant details that could help in future analysis?

5. This task requires you to apply analytical thinking to your documented flaws, tracing them back to their fundamental origins within the robot's system.

   a. **Flaw Review and Initial Analysis:** Review the complete list of flaws you documented in Task 1. Consider the robot's design, its programming, and its intended logic.

   b. **Root Cause Hypothesis:** For each identified shortcoming, propose its most likely **root cause. You must categorise each root cause as either an algorithm flaw, a design flaw, or a coding error.** Refer to your knowledge of robot components, control systems, and programming principles.

   c. **Justification of Trace:** Provide a detailed justification for why you believe the flaw falls into your chosen category. Explain how the observed behaviour connects to the proposed algorithmic logic, physical design, or code implementation.

   d. **Findings Documentation:** Compile your identified flaws, their proposed root causes, and your justifications into a structured technical document. This document should follow the reporting format for troubleshooting and error correction, as introduced in Year 2, Weeks 23 and 24.

6. This task challenges you to implement solutions for identified flaws and refine the robot's performance through a continuous, iterative cycle.

   a. **Flaw Selection for Correction:** From your analysis of shortcomings in Task 2, select at least one critical flaw that you will attempt to rectify, or one area where the robot's performance can be significantly improved.

   b. **Correction Implementation:** Based on your hypothesised root cause, implement a specific modification to the robot. This could involve:

      i. Adjusting the robot's algorithm (e.g., refining its decision-making logic or control parameters).

      ii. Modifying the robot's design (e.g., altering a physical component, adjusting sensor placement, or reinforcing a structural element).

      iii. Correcting an error in the robot's code (e.g., fixing a syntax error, adjusting variable assignments, or refining a loop).

7. **Re-Testing and Documentation:** After implementing your modification, re-run the relevant tests from Task 1. Observe carefully whether the flaw has been resolved and if the robot's performance has improved. Document the specific change you made, the detailed outcome of the re-test, and any new issues that may have emerged as a result of the correction.
8. **Reflection and Iteration:** Critically evaluate the results of your re-test. If the flaw persists, if performance has not improved as expected, or if new issues have appeared, reflect on the reasons. Propose further adjustments and repeat the "implement correction, re-test, document" cycle until the robot performs satisfactorily or reaches an optimal level for the given task. All iterations and their outcomes must be meticulously documented in the technical document format.

PEDAGOGICAL EXEMPLARS

This section offers guidance on how to administer the lesson, focusing on practical engagement and structured learning. The aim is to facilitate a deep understanding of robot troubleshooting and iterative improvement, with careful attention to differentiation and inclusivity for all learners.

1. **Step 1: Setting the Stage for Systematic Troubleshooting**

   Begin by reviewing Year 2 troubleshooting steps: routine establishment, problem isolation, observation, expectation setting, and system understanding. Introduce testing for fault detection and performance enhancement. Connect to real-world robotic challenges.

To ensure differentiation and inclusivity

   a. Use visual aids (e.g., troubleshooting flowchart) for visual learners.
   b. Facilitate a brief discussion for learners to share troubleshooting experiences, allowing verbal or non-verbal contributions.

2. **Step 2: Guiding Test Plan Development and Flaw Identification**

   Assign groups a functional robot and a task. Guide them to create a test plan, encouraging precise observation and documentation of unexpected behaviours.

To ensure differentiation and Inclusivity

   a. Provide a test plan template for structure.
   b. Challenge advanced learners with edge-case testing.
   c. Offer varied recording methods (written logs, checklists, photos/videos) and verbal/typed options for motor challenges.

3. **Step 3: Facilitating Root Cause Analysis and Categorisation**

Discuss algorithm flaws, design flaws, and coding errors with relatable examples. Instruct learners to categorise observed flaws using robot design details (e.g., pseudocode).

To ensure differentiation and Inclusivity

a. Supply a "Flaw Categorisation Guide" for clarity.

b. Foster group discussions to refine hypotheses, aiding collaborative learning.

4. **Step 4: Supporting Documentation**

   Guide groups to select one flaw for correction, providing tools for algorithm, design, or code adjustments. Emphasise iterative testing: modify, re-test, and document.

To ensure differentiation and inclusivity

a. Suggest simpler algorithm/design fixes for coding-challenged learners.

b. Provide clear tool instructions.

c. Support a "change log" for documentation, with adaptive tools or peer help for motor challenges.

5. **Step 5: Supporting Documentation**

   Collect technical documents and facilitate a class discussion on troubleshooting insights and lessons learned. Highlight the documentation's professional value.

To ensure differentiation and inclusivity

a. Provide a documentation checklist.

b. Allow reflections via writing, verbal summaries, or diagrams.

c. Recognise systematic thinking and effort, not just successful fixes.

KEY ASSESSMENT LEVEL 4: INDIVIDUAL PRACTICAL TASK

(*15 marks*) - (Suggested Modes: Individual Project Assessment)

This task assesses the learner's ability to systematically test a robotic system, analyse flaws, implement corrections, and document the process, reinforcing troubleshooting and iterative improvement skills in a practical, structured manner.

Project Title: Troubleshooting and Improving a Robotic System

Task: Evaluate a provided operational robotic system (physical or simulated) designed for a specific task (e.g., line-following, sorting, or obstacle avoidance). Develop and execute a test plan to identify flaws, trace their root causes, implement a correction for one critical flaw, and document the iterative process. Submit a concise technical report (maximum 2 A4 pages) detailing the process and findings, adhering to the troubleshooting documentation structure from Year 2, Weeks 23 and 24.

1. **Test Plan Development (4 marks):** Create a test plan including:

   a. Test objectives (e.g., evaluating task accuracy or speed).

   b. Test environment (e.g., lighting, surface type).

   c. Specific test cases with clear steps.

   - **d.** Expected outcomes for each test case.
   - **e.** Observation and documentation methods (e.g., visual inspection, sensor data).
2. **Flaw Identification and Root Cause Analysis (4 marks)**
   - **a.** Execute the test plan and document observed flaws (description, timing, frequency).
   - **b.** Categorise each flaw as an algorithm flaw, design flaw, or coding error, with justification.
3. **Correction and Iteration (4 marks)**
   - **a.** Select one critical flaw and implement a targeted fix (e.g., algorithm adjustment, design modification, or code correction).
   - **b.** Re-test the robot and document the results, noting resolution or new issues.
   - **c.** Reflect on the outcome and suggest further improvements if needed.
4. **Technical Report (*3 marks*):** Include:
   - **a.** Description of the robot and its task. Test plan summary (objectives, environment, test cases).
   - **b.** List of identified flaws with root cause analysis and categorisations.
   - **c.** Details of the correction, re-test results, and reflection on the iterative process.
   - **d.** Conclusion on lessons learned about troubleshooting and robot improvement.

   (***15 marks***)

Conclusion

The week's lesson on testing and refining operational robotic systems was carefully structured to enhance learners' troubleshooting proficiency. Emphasis was placed on crafting comprehensive test plans to evaluate robot performance, as exemplified in tasks like sorting or navigation. Flaws were systematically identified, categorised as algorithmic, design, or coding issues, and documented with precision. Through iterative testing and correction, learners honed skills in diagnosing root causes and implementing targeted fixes, fostering resilience and analytical thinking. The structured documentation process underscored the importance of methodical approaches, equipping learners to optimise robotic systems for real-world applications with reliability and efficiency.

PLC Session 7 & 8

45 minutes	Planning & Reviewing the Learning Plan

1. Begin the session by recapping lessons from the previous year/session/week, focusing on how learning was applied.
2. Use your subject specific app to explore the content before planning. Ask the app to
   - **a.** explain the week's concepts in a practical way that will make it easier for learners to understand

b. identify two or three common misconceptions learners may have on the week's content and how to address them using contexts familiar to Ghanaian learners. (NTS 3a, 3g and 3m).

3. Use your subject specific app to plan your lesson. Ask the app to

a. suggest the most effective method for teaching the week's concepts to learners with diverse backgrounds.

b. suggest learning activities, differentiation and essential questions based on the method of teaching (NTS 3a-3q)

c. review your prepared learning plan and address any challenging issues you may have identified in your subject area (NTS 3a-3c).

Note

1. Take note of the recap topics linked to Years 1 and 2 while preparing for your lesson.
2. Refer to the Values Learning Community Handbook and link the values for the week in your lesson to ensure that learners are applying what they learn.
3. Plan the week's key assessment with your app

a. The key assessment for the week is ***individual project task***. Chat with your app to plan this assessment, indicating what you and your learners will do before, during and after the task (NTS 3k-3q).

10 minutes	Content Exploration

4. As a subject group, select one problem or idea/term in the week's content that is challenging (not a routine procedural question, but one that requires conceptual understanding) and then

a. use the app, ask it to solve the problem step by step and explain the reasoning behind each step (you may be assisted by a teacher who is strong in the content area)

b. all teachers in the group work through a similar problem individually for 3–4 minutes

c. the group then works through the problem together step by step

d. highlight the most common errors made at each step and discuss (NTS 1c, 1e, 2c, 2e, 3e-3j).

Note

1. This part is not just about 'How do I teach this?' but how do I teach it for learners to understand?
2. If the group still shares the same content gap, use the app: ask it to solve the problem step by step and explain the reasoning behind each step.

3. Problems/ideas discussed should be sourced from the Teacher Manual content, not invented separately. This keeps the content exploration directly connected to what teachers will deliver.

30 minutes	Enactment

5. Enact a teaching activity based on your learning plan (NTS la-1b, 1e-lg, 2c, 2e, 2f, 3c-3j and 3o).
   a. Colleagues observe with specific attention to clarity of concept explanation, use of familiar Ghanaian examples and how misconceptions are handled if they arise.
   b. After the enactment, ask for structured feedback from the group
      i. was the concept explained in a way that a learner with no prior knowledge could follow?
      ii. were the examples grounded in contexts familiar to Ghanaian learners?
      iii. did any misconceptions surface and if so, how were they addressed?

5 minutes	Reflection

6. Reflect and share your views on the session. One thing that worked well, what did not work well and what you will do differently. (NTS 1a, 1b).
7. Identify a critical friend to observe and provide feedback on your lesson in relation to the PLC Session conducted (NTS 1f, 1g).
8. Remember to
   a. transfer your chats into your learning planner template.
   b. read your Teacher Manual and related Learner Material in preparation for the next session. Take note of the recap weeks, which are linked to Years 1 and 2 (NTS 3a).
   c. integrate National Values, GESI, SEL, and 21st century skills into your learning activities, assessment tasks, and rubrics/mark schemes.

Section Review

The section effectively deepened learners' understanding of autonomous robotic systems. Designing feedback-driven prototypes enhanced efficiency in local industries, as seen in applications like tomato sorting. Reverse-engineering exercises clarified how sensors and actuators enable autonomous behaviour, with documented control logic reinforcing practical insights. Troubleshooting lessons honed skills in creating test plans, identifying flaws, and implementing iterative fixes, ensuring reliability and performance. Through structured documentation and analytical approaches, learners developed competencies to innovate and optimise robotic solutions, equipping them to address real-world challenges with precision and creativity in diverse contexts.

ADDITIONAL READING

Resource	QR Code to Access Resource
1. LEGO Robotics Spike Prime Quality Check Bot (Video Resource).	
2. Fruit Sorting Robot (Video Resource).	
3. How Do I Troubleshoot a Robot at Competition? (Video Resource).	
4. Appyling the Troubleshooting Process (Video Resource)	

MARKING SCHEMES FOR THE KEY ASSESSMENTS

WEEK 4

Marking Scheme – Short Answer and Extended Response Essays

Question 1 – 7 marks	
1. Identified Variable and Justification (2 marks)	**1 mark –** Variable: Peel thickness / Amount of cassava flesh removed with peel / Depth of peeling blade.
	1 mark – Justification: Inconsistent peel thickness leads to significant waste of edible cassava or inefficient peeling if too much peel remains. Controlling this directly impacts material efficiency.
2. Sensor Type and Explanation (2 marks)	**1 mark –** Sensor Type: Ultrasonic sensor / Laser distance sensor / Optical sensor with depth perception.
	1 mark – Explanation: An ultrasonic/laser distance sensor can accurately measure the distance from the peeling blade to the cassava surface, providing data on peel thickness. An optical sensor could potentially identify remaining peel versus peeled flesh.
3. Feedback Control Logic Description: (3 marks)	**3 marks –** Logic: The sensor (e.g., ultrasonic) measures the current peel thickness. This measurement is compared to a desired set-point(optimal peel thickness). If the sensed thickness deviates from the set-point (error), the controller adjusts the actuator (e.g., a motor controlling the blade's depth or pressure) to correct the thickness, ensuring minimal waste. The system continuously monitors and adjusts. (3 marks for clear, sequential explanation of sensing, comparison, and actuation based on error)
Question 2 – 3 marks	
Explanation of Autonomous Operation and Example	**2 marks –** The feedback loop facilitates autonomous operation by enabling the system to continuously monitor its own output or environmental state (Sensed State) against a desired condition (Desired State). Any deviation (Error) triggers the Processor/Controller to command the Actuator to adjust the System/Environment, thereby closing the loop and maintaining the desired state without constant human input. This continuous self-correction allows for adaptation to changing conditions and sustained performance.
	1 mark – Description of a specific Ghanaian Industry Example (beyond tomato sorting) that illustrates the explanation.

QUESTION 2

Rubric - Individual Task

Criteria	Mark Distribution	
Problem Description (3 marks)	**0 Marks**	Unclear problem; no clear inefficiency identified.
	1 - 2 Marks	Problem identified: basic description of inefficiency.
	3 Marks	Clear and specific description of the local industry problem and its inefficiencies.
Proposed Prototype System Concept (4 marks)	**0 - 1 Marks**	Sketch/description unclear; components missing or inappropriate.
	2 - 3 Marks	Basic concept with some components; justifications are weak.
	4 Marks	Clear concept with appropriate key components identified and justified; sketch/diagram present.
Feedback Control Mechanism (4 marks)	**0 - 1 Marks**	Diagram missing/incorrect; logic unclear or flawed.
	2 - 3 Marks	The diagram presented but incomplete; the logic is basic or with minor errors.
	4 Marks	Accurate and well-labelled feedback loop diagram; clear, logical description of feedback control.
Autonomous Operation & Efficiency Improvement (2 marks)	**0 Marks**	Explanation absents or irrelevant.
	1 Mark	General explanation: weak link to autonomy or efficiency.
	2 Marks	Clear explanation of autonomous operation and specific, well-justified efficiency gains.
Operational Sequence Outline (2 marks)	**0 Marks**	No sequence or major errors.
	1 Mark	Basic sequence, difficult to follow or incomplete feedback loop.
	2 Marks	A clear and accurate flowchart or pseudocode effectively shows the feedback loop within the operational sequence.
Total	**15 Marks**	

WEEK 5 & 6

QUESTION 1

Marking Scheme - Short Answer and Extended Response Essays

Question 1 – 8 marks	
1. Primary Sensor and Actuator with Justification (**2 marks**)	**1 mark –** Sensor: Light sensor / Photoresistor (LDR). Justification: This sensor is needed to measure the ambient light levels, which are the variable triggering the door's autonomous action.
	1 mark – Actuator: Servo motor / DC motor with gear system. Justification: This actuator is needed to physically open and close the door, converting electrical signals into mechanical motion.
2. Core Feedback Loop Description (**3 marks**)	**1 mark –** Sensed State: The current ambient light level.
	1 mark – Desired State: Light level above threshold (for open) or below threshold (for closed).
	1 mark – General Action: The system senses the light level. If the light level is below the threshold for closing, the actuator closes the door. If it is above the threshold for opening, the actuator opens the door.
3. Control Logic (Pseudocode or Flowchart) (**3 marks**)	**3 marks –** Clearly drawn flowchart or pseudocode that illustrates the intended operation. Look out for the correct implementation of decision points for dusk and actions to open or close.
Question 2 – 2 marks	
Deducing Control Logic and Prior Knowledge Reliance (**2 marks**)	**1 mark –** Deducing the control logic relies on prior knowledge of feedback systems by understanding that the system uses the light sensor's input (sensed state) to determine the light bulb's output (actuator action) to achieve a desired state (on when dark, off when bright). This creates a closed loop where the output (light/dark) feeds back to influence the input.
	1 mark – It also relies on basic programming concepts such as conditional statements (e.g., IF-THEN-ELSE) and loops (LOOP FOREVER) to represent the logical rules the system follows. The light level acts as a condition that triggers specific actions.

Question 2
Rubric - Group-Based Task/Project

Criteria	Mark Distribution	
System Observation & Component ID **(5 marks)**	**0 - 1 Marks**	Superficial observation; components incorrectly identified.
	2 - 3 Marks	Basic observation: some components identified with a limited function description.
	4 - 5 Marks	Detailed observation of behaviour; accurate identification of key components and their probable roles.
Feedback Loop Dedication & Diagram **(5 marks)**	**0 Marks**	Loop not identified or diagram incorrect/unlabelled.
	2 - 3 Marks	Loop identified but with minor inaccuracies or incomplete labelling.
	4 - 5 Marks	Accurate deduction of the primary feedback loop; clear, correctly labelled block diagram.
Control Logic Inference & Representation **(5 marks)**	**0 Marks**	Logic not inferred or representation flawed.
	2 - 3 Marks	Basic logic inferred; representation has minor errors or is incomplete.
	4 - 5 Marks	Precise and logical inference of control logic; accurate and well-structured flowchart/ pseudocode.
Explanation & Justification **(5 marks)**	**0 Mark**	Explanation is unclear or lacks justification.
	2 - 3 Marks	Some explanation, justification for deductions is weak.
	4 - 5 Marks	Clear explanation of the system's autonomous function and logical justification for inferred controls.

Presentation Clarity & Learning Insights (**5 marks**)	**0 Marks**	Presentation unclear; no insights shared.
	2 - 3 Marks	Presentation generally clear; some insights shared.
	4 - 5 Marks	Highly clear and engaging presentation; deep insights shared about autonomous control design from the RE process.
Total	**25 Marks**	

WEEK 7 & 8

RUBRIC - INDIVIDUAL-PRACTICAL TASK

Criteria	Mark Distribution	
Test Plan Development (**4 marks**)	**0 –1 Mark**	A test plan is incomplete, vague, or lacks clear objectives, test cases, or expected outcomes.
	2 - 3 Marks	The test plan includes objectives, test cases, and outcomes, but lacks detail or clarity in environment or observation methods.
	4 Marks	Comprehensive test plan with clear objectives, detailed test cases, defined environment, expected outcomes, and precise observation methods.
Flaw Identification & Root Cause Analysis (**4 marks**)	**0 Marks**	Flaws are poorly described or not identified. Root causes are absent or incorrectly categorised.
	2 - 3 Marks	Flaws are identified with basic descriptions. Root causes categorised with some justification but lack depth.
	4 Marks	Flaws are clearly described (behaviour, timing, frequency). Root causes accurately categorised (algorithm/design/coding) with detailed justifications.

Correction & Iteration (**4 marks**)	**0 - 1 Mark**	No correction attempted or irrelevant. Re-test results were undocumented or unclear.
	2 - 3 Marks	Correction attempted but partially effective. Re-test results documented with minor gaps; limited reflection on outcomes.
	4 - 5 Marks	Targeted correction implemented for one flaw. Re-test results thoroughly documented, with clear reflection on resolution and potential further improvements.
Technical Report (**3 marks**)	**0 Marks**	The report is disorganised, missing key sections, or unclear.
	1- 2 Marks	The report follows structure but lacks clarity or detail in some sections (e.g., test plan, flaws, or iteration).
	3 Marks	Well-organised, clear, and detailed report covering robot description, test plan, flaws, corrections, re-test results, and reflective conclusion.
Total	**15 Marks**	

SECTION 3: SENSORS AND ACTUATORS 3

STRAND: PRINCIPLES OF ROBOTIC SYSTEMS

Sub-Strand: Sensors and Actuators

Learning Outcome: *Observe and explain the underlying STEM concepts of functional automated system processes and recommend feasible automation designs for unautomated industrial processes.*

Content Standard: *Demonstrate analytical skills in the selection of final control elements for industrial control solutions.*

INTRODUCTION AND SECTION SUMMARY

This section builds on Year 1 and 2 knowledge of basic sensors and actuators, and control principles from Section 2, to focus on advanced industrial actuators and analysing functional automated systems. Learners will move beyond component types to observe and explain the STEM concepts in real or simulated industrial processes. The core objective is to develop skills in analysing unautomated industrial tasks and recommending feasible automation designs. A key emphasis is placed on the strategic selection and justification of appropriate advanced industrial actuators, known as final control elements, for specific industrial control requirements. By the end, learners will enhance their analytical abilities in system observation and demonstrate justified selection of actuators for industrial automation solutions, preparing them for practical application in local contexts.

The week and indicators covered by the section are:

Week 9

- Analyse the operating principles of advanced industry-based actuators (valves, hydraulic-based actuators, contactors, relays, etc.)
- Evaluate an industrial control requirement and recommend final control actuators with appropriate justification.

SUMMARY OF PEDAGOGICAL EXEMPLARS

Teaching in this section adopts an analytical and application-focused approach using industrial examples. Strategies include analysing case studies or observing functional automated systems (via videos, simulations, or diagrams) to identify components and understand underlying STEM principles. Learners will evaluate industrial control needs based on scenarios and recommend appropriate advanced actuators (valves, hydraulics, etc.), justifying their choices based on requirements and local context. These activities will involve discussions, group work, and research. Differentiation is crucial, adjusting the complexity of systems analysed and requirements evaluated. Support can include structured analysis templates, while gifted learners can explore complex systems or uncommon actuators. Accessibility is ensured through clear visuals, simple language for technical terms, and diverse methods for demonstrating understanding and recommendations.

ASSESSMENT SUMMARY

The following table presents a sample of the key assessments used in this section.

Indicators	DOK Level	Assessment Strategy and/or Mode
1. Analyse the operating principles of advanced industry-based actuators (valves, hydraulic-based actuators, contactors, relays, etc.)	1 & 2	Short Answer Questions and Extended Response Essays – (Suggested Modes: Individual/Group Presentation, Mid-semester Assessment, End-of-semester Assessment)
2. Evaluate an industrial control requirement and recommend final control actuators with appropriate justification.	4	Short answer questions and a Group-Based Task/Project – (Suggested Modes: Individual/Group project/research/case study)

WEEK 9

Learning Indicators

1. *Analyse the operating principles of advanced industry-based actuators (valves, hydraulic-based actuators, contactors, relays, etc.)*
2. *Evaluate an industrial control requirement and recommend final control actuators with appropriate justification*

FOCAL AREA 1: ANALYSING INDUSTRIAL ACTUATOR PRINCIPLES

Introduction

This lesson introduces learners to advanced actuators commonly employed in industrial automation, moving beyond the basic motors and servos typically used in smaller robots. It examines the fundamental principles governing the operation of components such as various valves, hydraulic systems, contactors, and relays. Understanding how these powerful final control elements function is essential for comprehending industrial automated processes. The lesson provides foundational knowledge necessary for analysing functional systems and recommending suitable automation solutions.

Building on Foundational Concepts

This focal area builds upon foundational knowledge of robot components from Year 1 and control systems from Year 2. Learners recall their understanding of basic actuators (Year 1, Weeks 3 & 4) and the scientific principles underlying component operation (Year 1, Week 6). They apply this by analysing how advanced industrial actuators function. Understanding how these components convert energy into physical action, a core concept from earlier years, is essential for grasping the operating principles of industrial valves, hydraulics, contactors, and relays discussed this week.

Recap Weeks

Year 1	**Week 3** 1. Describe the attributes and functionalities of a robot's subsystems and how they interconnect **Week 6** 1. Discuss the scientific principles underlying the operation of sensors. (Provides a parallel to understanding actuator principles). **Week 9** Identify the components of an electronic circuit and their functions. **Weeks 19 & 20** Sit in groups and create robots using robotic kits and/or local materials to implement basic mechanics for actuations that make use of the following: GEARS, VEHICLES, ARMS, WINGS & OTHERS

2	**Weeks 19 & 20** Learners should create robots using robotic kits and/or local materials to implement basic mechanics for actuations that make use of the following: GEARS, VEHICLES, ARMS, WINGS & OTHERS

Content Knowledge

Understanding Actuators as Final Control Elements

In automated systems, actuators serve as the ‘muscles’ or ‘doers’, directly influencing the physical process being controlled based on signals from a control system. While learners have worked with basic actuators like small DC motors and servos in their senior high school robotics kits (components they are familiar with from Year 1, e.g., LEGO Education SPIKE Prime motors), industrial automation frequently requires components capable of handling significantly higher forces, pressures, voltages, or flow rates. These advanced actuators, often called ‘final control elements’ perform specific, powerful functions necessary for large-scale operations, such as precise movement of heavy loads, regulating the flow of liquids or gases, or safely switching high-power electrical equipment.

Categories of Advanced Industrial Actuators

Industrial actuators can be broadly categorised by their energy source or primary function. This lesson focuses on understanding the operating principles of several key types:

1. **Valves:** Valves regulate the flow or pressure of fluids (liquids or gases) in piping systems, enabling precise control in industrial processes. They can start, stop, or adjust fluid flow, similar to how a tap controls water.

Figure 9.1: *Industrial Valve*

   a. **Operating Principle:** Valves work by physically altering the size of the flow path or blocking it entirely. Automated valves use an external signal (electrical, pneumatic, or hydraulic) to move an internal component (like a disc, ball, or gate) that regulates the fluid flow. For instance, a solenoid valve uses an electrical coil to create a magnetic field, which then moves a piston or plunger to open or close the valve. This principle is directly comparable to how a low-voltage electromagnet is used as an end-effector (gripper) on an SHS robotic arm to pick and place metallic materials, or how an electromagnetic latch is used to actuate a sorting gate on a classroom recycling conveyor system, but scaled up in this context to mechanically regulate high-pressure fluid flow..

Types: ***The following are some common types of valves:***

i. **Gate Valve:** Linear motion valve for starting/stopping flow; used in oil and gas industries.

ii. **Globe Valve:** Regulates flow with a movable disc; common in water cooling systems.

iii. **Ball Valve:** Rotational valve with a ball-shaped disc; cost-effective for chemical storage.

iv. **Butterfly Valve:** Lightweight rotary valve for slurries; used in food processing.

v. **Solenoid Valve:** Electrically controlled for fast response; ideal for automated systems.

b. **Relevance to Robotics:** Valves mirror the control logic in robotics, where a signal triggers a specific action, scaled for industrial fluid management.

c. **Applications:** Essential in Ghanaian industries like cocoa processing (fluid control in chocolate production), water treatment, and energy plants.

For more information on the types of valves, their operating principles, as well as their relevance in industries, click the link or scan the QR code below:

Link and Description	QR Code
Types of Valves | All in One Guide to Industrial Valve Types	

2. **Hydraulic-Based Actuators:** Hydraulic cylinders produce linear motion by converting hydraulic fluid pressure into mechanical force, ideal for tasks requiring significant strength, such as lifting or pressing.

Figure 9.2: *Hydraulic Actuator*

a. **Operating Principle:** Based on Pascal's principle, hydraulic systems transmit pressure through a fluid to exert force on a piston within a cylinder. A small force applied over a small area can generate a much larger force over a larger area. Hydraulic cylinders provide linear motion, while hydraulic motors provide rotary motion. Control is typically achieved by directing the flow of pressurised fluid using valves. The concept of generating motion or force from an energy source is akin to how an electric motor spins when electricity is applied, but hydraulic systems achieve far greater force, comparable to the strength needed to lift parts in heavy machinery used in mining or construction.

b. **Types:** The following are some common types of hydraulic cylinders:

   i. ***Single-Acting Cylinder:*** Fluid pressure extends the rod; a spring retracts it; used in simple lifts.

   ii. ***Double-Acting Cylinder:*** Fluid pressure extends and retracts the rod; suited for heavy machinery.

   iii. ***Telescopic Cylinder:*** Extends in stages for long strokes in compact spaces; common in forklifts.

   iv. ***Tie-Rod Cylinder:*** Robust for heavy-duty tasks; used in construction equipment.

c. **Relevance to Robotics:** Hydraulic cylinders are themselves direct fluid-driven actuators used in automated and robotic systems where extreme force is required. They demonstrate how controlled fluid energy is converted into high-force linear mechanical motion, which is a fundamental principle governing high-payload industrial arms, heavy-duty mobile robots, and automated manufacturing systems.

d. **Applications:** Widely used in heavy industries where high force or torque is required, such as in construction equipment, manufacturing presses, and agricultural machinery. Vital in Ghana's mining (excavators), agriculture (tractors), and construction (cranes) sectors.

The video below explains the different types of hydraulic cylinders, their functions and working principles.

Link and Description	QR Code
Functions & Types of Hydraulic Cylinders	

3. **Contactors:** Contactors are heavy-duty switches that control high-power electrical circuits, such as motors or lighting, using a low-voltage signal, ensuring safe operation of large loads.

Figure 9.3: *Contractor*

a. **Operating Principle:** A contactor uses a low-voltage control signal to energise an electromagnetic coil. When the coil is energised, it creates a magnetic field that pulls a movable armature, causing a set of heavy-duty electrical contacts in the high-power circuit to close (or open). When the control signal is removed, a spring typically returns the contacts to their original state, interrupting the high-power circuit. This is analogous to a small relay sometimes used in SHS electronics or robotics projects to switch a small light or motor using a low-voltage microcontroller signal, but contactors are built to handle much higher voltages and currents safely, essential for industrial machinery.

b. **Types:** The following are some common types of contactors:

   i. **Electrically Held:** Contacts revert when power is removed; used in temporary lighting control.

   ii. **Mechanically Held (Latching):** Contacts remain in position without continuous power; ideal for stable circuit states.

   iii. **Normally Open/Normally Closed:** Configurable based on circuit needs; used in motor control.

   iv. **Relevance to Robotics:** Contactors and electrical control systems are fundamental to robotics and industrial automation, enabling low-power robotic controllers to safely switch high-power circuits and regulate the operation of heavy motors, high-draw actuators, and large industrial components. These systems are widely applied across Ghanaian manufacturing companies (for heavy machine sequencing), processing plants (for automated pump control), and utility infrastructure.

   v. **Applications:** Contactors extend the concept of switching in robotics to manage high-power industrial systems safely.

The video below explains what a contactor is and its working principles as well as its practical applications.

Link and Description	QR Code
What is a Contactor and How Does it Work?	

4. **Relays**: These are electrically operated switches. They use an electromagnet to operate a switch mechanically, often used to control one electrical circuit by opening or closing contacts in another circuit.

Figure 9.4: *Relays*

a. **Operating Principle:** Like contactors, relays use an electromagnetic coil activated by a control voltage. Energising the coil creates a magnetic field that moves an armature, changing the state of one or more sets of electrical contacts. The key difference from contactors lies in their power handling capacity. Relays are designed for low to medium current applications, typically handling 5–10 amps. They control smaller loads, such as sensors, fans, or lights, in automation systems. Contactors on the other hand are built for high-power circuits, managing 30–40 amps or more. They switch heavy loads like large motors, industrial lighting, or heating systems. Looking at their designs, relays are usually small, lightweight and less rugged whereas contactors are large, robust and heavy-duty in nature.

b. **Types:** The following are some common types of relays:

   i. Normally Open (NO) relay: In its de-energised state (no power to the coil), the relay's contacts are open, meaning the secondary circuit is incomplete, and no electricity flows to the load (e.g., a fan or light is off). When the coil is energised, the contacts close, completing the circuit and activating the load.

   ii. Solid-State Relay (SSR): A solid-state relay is an electronic switching device with no moving parts, using semiconductor components to control circuits. Instead of an electromagnetic coil, an SSR employs an LED on the primary side to emit light, which activates a photosensitive transistor on the secondary side, allowing current to flow.

c. **Relevance to Robotics:** Relays allow low-power control signals from a robot's microcontroller (e.g., 5V DC) to safely switch higher-power loads like motors, pumps, and solenoids while providing electrical isolation to protect sensitive control electronics.

d. **Applications:** Used in Ghanaian automation systems (e.g., irrigation control), building management (HVAC), and small-scale industrial logic circuits.

The video below explains what a relay is and its working principles as well as its practical applications.

Link and Description	QR Code
Understanding Relays and Relay Logic in Industrial Automation	

e. **Analysing Operating Principles:** The table below summarises the actuators' operating principles, highlighting energy conversion and application contexts:

Feature	Valves	Hydraulic Actuators	Contactors	Relays
Energy Type	Electrical, pneumatic, hydraulic	Hydraulic	Electrical	Electrical
Control Signal	Electric current or pressure	Hydraulic pressure	Low-voltage electric	Low-voltage electric
Output Action	Fluid flow change	Linear or rotary motion	Circuit switching	Circuit switching
Application Environment	Process industries	Heavy-duty machinery	Electrical distribution	Control logic and low-current switching
Power Handling	Medium–High	Very High	High	Low–Medium
Response Time	Fast (esp. solenoid)	Moderate	Moderate–Fast	Fast

Learning Tasks

The following task guides learners to analyse the operating principles of advanced industry-based actuators. Depending on the available time or resources, administer one or more of the following learning tasks to help learners reinforce understanding and acquire new knowledge or skills.

1. For each assigned actuator (e.g., a type of valve, a hydraulic cylinder, a contractor, a relay), research and analyse its operating principle. (DoK 1 & 2)
   a. Identify the energy source it uses.
   b. Describe how the input control signal (e.g., electrical voltage, air pressure) causes the actuator to perform its function.
   c. Explain the internal mechanism that translates the input signal into physical action or control.
   d. State the primary function the actuator performs in an automated system.

e. Identify common applications of this actuator in industrial settings, particularly in local industries if examples can be found.

Document your analysis for each actuator using diagrams and written descriptions. Prepare to share your findings with the class.

Hint

- *The specific actuators assigned or chosen should align with available resources, such as diagrams, videos, or physical examples if accessible. Encourage learners to use diagrams to illustrate the internal workings.).*

2. Select two distinct types of industrial actuators (e.g., a valve and a hydraulic actuator, or a contactor and a relay). Construct a comparison that highlights their differences and similarities in terms of:
   a. Energy type used.
   b. Output action.
   c. Typical power handling capacity.
   d. Response time.

Document your comparisons for each actuator using diagrams and written descriptions. Prepare to share your findings with the class. (DoK 1 & 2)

3. *Extension Question:* For a given industrial scenario (e.g., controlling a large pump in a water treatment plant, lifting heavy machinery in a factory, or automating a high-speed packaging line), justify which specific actuator type would be most appropriate. Explain your reasoning based on the actuator's operating principle, power handling, and typical applications. (DoK 3)

PEDAGOGICAL EXEMPLARS

This focal area provides guidance for the teacher on facilitating the analysis of advanced industrial actuator principles, incorporating differentiated approaches. The aim is to build on learners' foundational knowledge of robotics by connecting it to the robust actuators used in industrial automation. Recognise that learners possess shared characteristics while also having individual differences in interests, readiness, and learning styles. Plan to address these variations. Provide multiple avenues for learners to engage with the lesson content, adjusting the level of detail or support as needed to help all learners achieve the lesson outcomes. Employ strategies such as Talk for Learning to encourage participation. Learners work in pairs or small groups to explain how different actuators operate, using diagrams or short video clips as prompts.

1. Step 1: Introduction to Industrial Actuators

a. Begin the lesson by revisiting the concept of actuators from Year 1 (Refer to Year 1 Manual, Section 3), reminding learners that these components are the 'movers' or 'doers' of a robotic or automated system. Introduce the idea that industrial settings require actuators capable of handling significantly greater forces, pressures, or electrical loads than those used in small robots.

b. Introduce the term 'final control element,' explaining its role in directly influencing a physical process. Present visual examples (via images, short videos, or diagrams) of industrial processes where advanced actuators are used (e.g., a robotic arm lifting heavy parts in a factory, automated valves controlling liquid in a processing plant, a large motor being switched on by a contactor).

c. Introduce the specific types of advanced actuators learners will analyse in this lesson: valves, hydraulic-based actuators, contactors, and relays.

To ensure differentiation and inclusivity:

- Provide simplified explanations and visual analogies for the operating principles of each actuator type for learners who may find the concepts challenging (e.g., comparing a valve to a tap controlling water flow, a contactor to a heavy-duty light switch operated remotely).
- For learners who grasp concepts quickly, briefly mention additional types of industrial actuators (e.g., pneumatic actuators, stepper motors for precision control) or prompt them to consider specific subtypes.
- Ensure all visual materials (images, diagrams, videos) are clear, well-labelled, and accessible to learners with varying visual abilities. Provide a concise glossary of new technical terms related to industrial actuators.

2. Step 2: Guiding Actuator Identification and Principle Application

a. Divide learners into small groups, ensuring a mix of abilities within each group. Assign or allow groups to choose specific actuators to analyse, based on available resources (physical examples, detailed diagrams, videos, online resources).

b. Guide groups as they research and analyse the operating principles of their assigned actuators. Circulate among groups, prompting their analysis. Ask targeted questions such as: "What energy source drives this actuator?", "How does the input control signal actually cause the physical action inside?", "What is the primary function this actuator performs in a system?"

c. Encourage groups to draw simple diagrams of the internal workings to help them understand and explain the principles.

To ensure differentiation and inclusivity

- For groups needing more support, provide partially completed diagrams or analysis templates with guiding questions for each actuator, focusing their analysis on identifying the energy source and the basic function.
- Challenge more advanced groups to research real-world specifications (e.g., typical voltage ratings for contactors, force output for hydraulic cylinders) or common failure modes for their assigned actuators.

- Provide large paper, whiteboards, or digital drawing tools for collaborative diagram creation. Allow learners to verbally explain their understanding to peers before formal documentation for those who prefer auditory processing. Ensure resources are available in various formats where possible.

3. **Step 3: Facilitating Comparative Actuator Analysis**

   a. Direct groups to select two distinct actuators for comparison, as outlined in Task 2. Guide them in constructing a structured comparison, focusing on key features such as energy type, output action, power handling, and response time.

   b. Following this, present several hypothetical industrial scenarios (e.g., opening a large dam gate, switching a heavy industrial motor, precisely controlling a chemical flow).

   c. Challenge groups to justify which specific actuator type would be most appropriate for each scenario, explaining their reasoning based on the actuator's unique operating principle, capabilities, and typical applications, especially in local industries.

To ensure differentiation and inclusivity

- Provide a comparative table template for groups to fill in, which can simplify the organisation of their comparison points.
- For learners requiring more support, suggest comparing actuators with very clear distinctions (e.g., relay vs. hydraulic actuator). For advanced learners, present scenarios with more nuanced requirements where multiple actuators might seem suitable, prompting deeper critical analysis.
- Encourage learners to refer back to the summary table provided in the knowledge content as a starting point for their comparisons. Promote peer discussion to help clarify concepts and consolidate understanding.

4. **Step 4: Guiding Real-World Application and Documentation**

   a. Provide groups with access to case studies, videos, or simplified simulations of industrial processes (e.g., a beverage bottling plant, a mining operation, a textile factory).

   b. Guide them to identify various actuators within these systems and describe their specific contributions to the overall process.

   c. Emphasise clear and concise documentation of their findings, including, optionally, simple diagrams illustrating actuator integration.

To ensure differentiation and inclusivity

- For learners who may find abstract industrial processes challenging, use highly visual resources (e.g., short, engaging industrial documentaries or animated simulations). Focus their identification on clearly visible or audibly active actuators.
- Challenge more advanced learners to consider the interaction between multiple actuators within a complex sub-system or to propose how an actuator's failure might impact the entire industrial process.

- Allow for varied documentation methods, including bullet points, brief paragraphs, or annotated diagrams. Provide opportunities for learners to verbally present their findings before formal written submission for those who learn best through verbalisation. Ensure digital resources are accessible and compatible with assistive technologies.

FOCAL AREA 2: SELECTING FINAL CONTROL ACTUATORS

Introduction

This second part of week 9 focuses on applying the understanding of industrial actuators to real-world automation challenges. Learners will examine the typical operation processes of any unautomated local industry. Learners should produce sensor and actuation requirement specifications for automating the processes involved in the operations of the local industry. Learners must recommend and justify actuator combinations that can be used to replace the manual processes of the local industry and sketch actuation designs for the automation of the processes identified. Learners will evaluate specific industrial control requirements, such as moving heavy loads, precisely controlling fluid flow, or safely switching high-power electrical equipment. Building on their analysis of actuator operating principles, the lesson guides learners in recommending the most appropriate final control actuators for a given unautomated task and providing clear justification for their selection based on the actuator's capabilities, the requirements of the task, and the context of a local industry.

Building on Foundational Concepts

Building on prior lessons, this second focal area requires learners to synthesise knowledge from multiple areas. It connects the analysis of actuator principles from Focal Area 1 to the evaluation of industrial control requirements. Learners apply their understanding of control system needs (Year 2, Section 2; Year 3, Section 2) and problem analysis skills (Year 3, Section 1) to determine actuator suitability. Recalling the capabilities of basic actuators from Year 1 informs their selection process for advanced components based on specific industrial needs and constraints.

Recap Weeks

Year 1	**Week 3** • Describe the attributes and functionalities of a robot's subsystems and how they interconnect (Focus on Actuation Subsystem). **Week 4** • Examine sensor power source inputs and outputs and classify them based on type. Apply mathematical knowledge for sensor data manipulation and digital control of actuators. (Focus on Actuators – power, inputs/outputs, digital control).

Year 2	**Section 2 (Weeks 3 & 4)** • Convert automated solution narratives into combinational circuit-controlled feedback and non-feedback design components. (Focus on understanding control requirements).
Year 3	**Section 2 (Week 4)** • Design a prototype feedback-driven autonomous system for efficiency improvement in an identified local industry. (Focus on applying control principles to real-world tasks and designing systems).

CONTENT KNOWLEDGE

Evaluating Industrial Control Requirements

Selecting the right actuator for an industrial automation task is critical for system performance, safety, and cost-effectiveness. This requires carefully evaluating the specific requirements of the control task, which often include:

1. **Type of Motion/Action:** Does the task require linear motion (pushing/pulling in presses or gates), rotary motion (e.g., turning valves or rotating conveyors), flow control (e.g., regulating liquids or gases), or electrical switching (e.g., turning high-voltage power on/off?)
2. **Force or Torque Required:** How much force (for linear motion) or torque (for rotary motion) is needed to perform the task? High-force applications may require hydraulic actuators, while Medium-force, fast operations may suit pneumatics. Low to moderate-force, precise applications may be handled by electric actuators.
3. **Speed of Operation:** How quickly does the actuator need to complete its action? Pneumatic and electric actuators are generally faster. Hydraulic actuators are slower but stronger.
4. **Precision and Accuracy:** How precise does the motion or control need to be? (While advanced precision actuators like stepper or servo motors exist, this lesson focuses on the provided types; however, the concept of precision is applicable). For valves, this might relate to how precisely the flow rate can be controlled.
5. **Operating Environment:** What are the conditions where the actuator will operate (temperature, humidity, presence of dust, corrosive substances, potential for explosions)? This influences the type of actuator and its required protection.
6. **Power Availability:** What type of power is available (electrical voltage/current, compressed air, hydraulic pressure)?
7. **Safety Requirements:** Are there specific safety considerations? For example, switching high voltage requires isolation, or systems might need to fail in a safe state (e.g., a valve closing automatically if power is lost).
8. **Cost:** What is the budget for the actuator? Industrial actuators can vary significantly in cost.

9. **Maintenance:** How complex is the actuator to maintain? What are the expected maintenance needs in a local context?

Matching Actuator Capabilities to Requirements:

Actuator Type	Operating Principle	Best For	Limitations
Hydraulic Actuator	Fluid pressure creates motion	High-force lifting, pressing	High cost, slower, complex, risk of leaks
Pneumatic Actuator	Compressed air moves piston/rotor	Fast, repetitive, moderate force tasks	Noisy, limited precision
Electric Actuator (DC/AC Motor)	Electrical energy drives motion	Controlled rotation, conveyors	May overheat with prolonged use
Servo Motor	Motor with feedback system	Precision control, robotics	Higher cost
Solenoid	Electromagnetic motion	On/off pushing or pulling	Limited to short stroke & force

a. **Valves:** Essential for controlling fluid flow and pressure. Different valve types offer varying levels of flow control precision.

b. **Contactors:** Ideal for safely switching high-current electrical loads, particularly inductive loads like motors.

c. **Relays:** Useful for lower-power switching, signal isolation, or implementing simple control logic by connecting contacts in different ways.

Recommending and Justifying Actuators

The task requires learners to recommend a *final control actuator* (or possibly a combination) for a given industrial control requirement and provide a justification. This justification should explicitly link the chosen actuator's operating principle and capabilities to the specific requirements of the task and the constraints of the industrial environment (ideally referencing local industrial contexts where applicable).

Justification Example

Task: A metal workshop lifts heavy metal beams using a hand operated system of chains and pulleys. The workshop requires this to be replaced by a safer and faster automated system to lift and hold steel beams weighing over 3 tonnes.

Recommendation: A **hydraulic actuator** (cylinder)

Justification: Hydraulic actuators generate high linear forces through pressurised fluid, making them ideal for safe heavy-duty lifting. This solution is already common in local metal fabrication industries, and suitable hydraulic systems are serviceable with available expertise.

This process requires learners to synthesise their understanding of actuator principles with the analysis of specific industrial needs, demonstrating analytical and evaluative skills.

Learning Tasks

The following task guides learners to evaluate industrial control requirements and recommend appropriate final control actuators with justification. Depending on the available time or resources, administer one or more of the following learning tasks to help learners reinforce understanding and acquire new knowledge or skills.

1. This task requires you to analyse a given industrial control scenario and propose suitable actuator types based on specific operational requirements. (DoK 4)
   a. **Scenario Analysis:** You will be provided with a detailed description of an industrial task requiring automated control (e.g., a specific step in a food processing line, a function in a mining operation, or a part of a water pumping station).
   b. **Requirement Identification**: From the scenario, identify and list at least four key control requirements. These requirements may include:
      i. The specific type of motion or action needed (linear, rotary, flow control, electrical switching).
      ii. The approximate force or torque required.
      iii. The necessary speed of operation.
      iv. Any relevant environmental conditions (e.g., dusty, high temperature, presence of liquids).
      v. Specific safety considerations.

2. **Actuator Proposal:** For each identified requirement, propose one or more types of industrial actuators (from the categories discussed: valves, hydraulic-based actuators, contactors, or relays) that you believe would be suitable. Briefly state why each proposed actuator meets the specific requirement.

3. Task requires you to justify your actuator choices based on their capabilities and acknowledge any potential limitations. (DoK 4)
   a. **Justification of Choice:** Select the single most appropriate actuator (or combination of actuators) to perform the entire industrial task from Task 1. Provide a comprehensive justification for your selection. Your justification must explicitly link the chosen actuator's operating principle, capabilities, and advantages directly to the specific requirements of the task.
   b. **Addressing Limitations:** For your chosen actuator(s), identify at least two potential limitations or challenges it might present in the given industrial scenario (e.g., high cost, noise, precision issues, specific maintenance needs, environmental vulnerability). Suggest how these limitations might be mitigated or managed.
   c. **Local Context Consideration:** If the scenario has a connection to a local Ghanaian industry, incorporate specific considerations relevant to that context in your justification

or limitation assessment (e.g., availability of maintenance expertise, common environmental factors, typical power supply conditions).

Hint

- *Provide varied scenarios that clearly highlight the need for different types of actuators. Examples: A system to control the flow of water into a tank; a machine needing to crush solid material; switching on a large electric pump; isolating a low-voltage control circuit from a high-voltage motor.).*

PEDAGOGICAL EXEMPLARS

This focal area offers guidance on how to facilitate learners' understanding of selecting the correct industrial actuators for specific tasks using scenarios and a project-based learning approach. The aim of this second focal area is to enable learners to critically evaluate industrial control requirements and justify their actuator choices, connecting theoretical knowledge with practical applications. This approach acknowledges that learners have diverse strengths and needs, ensuring multiple avenues for engagement and comprehension.

1. **Step 1: Introduction to Actuator Selection Criteria**

   Begin by reminding learners that choosing the right "muscle" for a robot or automated system is crucial. Introduce the various factors that influence actuator selection: the type of motion needed, the force or torque required, operational speed, precision, environmental conditions, power availability, safety needs, cost, and maintenance. Use everyday examples to make these criteria relatable before applying them to industrial contexts (e.g., choosing a bicycle pump versus a car jack for lifting).

To ensure differentiation and inclusivity:

- Provide a visual checklist or graphic organiser displaying the actuator selection criteria, assisting visual learners and those who benefit from structured information.
- Initiate a brief "Think-Pair-Share" activity where learners discuss which criteria might be most important for a simple, familiar task (e.g., automatically opening a classroom door), promoting active participation.

2. **Step 2: Facilitating Requirements Analysis and Actuator Proposal**

   To guide learners in analysing industrial scenarios and proposing suitable actuator types based on specific requirements, provide each group with a detailed description of an industrial task requiring automated control. These scenarios should clearly present various operational requirements. Guide learners as they carefully read and identify the key control requirements from their assigned scenario. Circulate among groups, prompting them to link identified requirements to the capabilities of different actuator types discussed in the previous lesson (valves, hydraulic, contactors, relays). Encourage them to propose one or more suitable actuators for each specific requirement.

To ensure differentiation and inclusivity:

- Offer scenarios with varying levels of complexity. Some learners may receive scenarios with very explicit requirements, while others may work with more open-ended situations that require deeper inference.
- Provide a structured worksheet with guiding questions to help learners systematically break down the scenario and list requirements.
- Encourage the use of colour-coding or highlighting to visually connect requirements with proposed actuator types, aiding organisation and comprehension.

3. **Step 3: Guiding Actuator Justification and Limitation Assessment**

   To assist learners in justifying their actuator choices and acknowledging potential limitations (Learning Task 2), direct groups to select the single most appropriate actuator (or combination) for their entire industrial task from Task 1. Guide them in developing a comprehensive justification, ensuring they explicitly link the chosen actuator's operating principle, capabilities, and advantages to the task's specific requirements. Prompt learners to also identify potential limitations or challenges their chosen actuator might present within the scenario and to suggest mitigation strategies. Encourage consideration of relevant local Ghanaian industrial contexts, if applicable, when discussing cost, maintenance, or environmental factors.

To ensure differentiation and inclusivity

- Provide sentence starters or structured paragraphs for learners who need support in articulating their justifications clearly.
- Challenge advanced learners to research specific technical specifications for their chosen actuators (e.g., IP ratings for environmental protection) to strengthen their justifications.
- Allow for varied presentation of justifications, such as brief verbal explanations accompanied by diagrams, or written reports, catering to diverse communication styles.

4. **Facilitating Class Discussion and Consolidation**

   To consolidate learning through class discussion and highlight key insights on actuator selection, bring the class together for a discussion on their actuator selections and justifications. Facilitate a dialogue that encourages learners to share their reasoning, discuss challenges faced, and compare different approaches to similar problems. Emphasise the importance of a systematic approach to evaluating requirements and the trade-offs involved in selecting industrial actuators. Reinforce the real-world impact of these decisions on system performance, safety, and operational efficiency in various industries within Ghana.

To ensure differentiation and inclusivity

- Establish clear ground rules for respectful questioning and constructive feedback during discussions.

- Ensure that all learners, including those who may be hesitant to speak, have an opportunity to contribute, perhaps through a structured "pass-the-mic" or written submission of one key takeaway.
- Summarise the main learning points on a whiteboard or projector, creating a visual reference for all learners.

KEY ASSESSMENT LEVEL 4: PROJECT

1. **Short answer questions (*Total 14 marks*)**

   DoK 1 (*2 marks each*)

   a. What is the primary role of a solenoid-operated valve in industrial systems?

   b. Name one key difference between a hydraulic actuator and a pneumatic actuator.

   c. What type of electromagnetic component is essential to the operation of an electromechanical relay?

   DoK 1 & 2 (*4 marks each*)

   a. Explain how a contactor differs from a standard relay in both construction and application within high-power systems.

   b. Describe the working principle of a hydraulic actuator and explain why it's suitable for high-force applications.

   DoK 4 Extended answer questions (*10 marks each*)

1. Consider a local food processing plant that needs to automate the control of steam flow into a cooking vat to maintain a precise temperature. (total 10 marks)

   a. Identify the specific industrial control requirement in this scenario. (*3 marks*)

   b. Recommend a suitable final control actuator (from the types discussed: valves, hydraulic actuators, contactors, relays) to manage the steam flow. (*2 marks*)

   c. Justify your recommendation by explaining why this actuator is appropriate for controlling fluid flow and maintaining temperature in this industrial context, referencing its operating principle. (Approx. 100-150 words) (*5 marks*)

1. A small local workshop uses a large electric motor to power a cutting machine. They want to automate the process of turning the motor on and off from a central control panel. (total 10 marks)

   a. Identify the specific industrial control requirement in this scenario. (3 marks)

   b. Recommend a suitable final control actuator to safely switch the power to this large electric motor. (2 marks)

   c. Justify your recommendation by explaining why this actuator is appropriate for handling high electrical loads and providing electrical isolation, referencing its operating principle. (Approx. 100-150 words) (5 marks)

2. Compare the typical applications and key advantages of hydraulic actuators versus contactors in industrial automation. Explain how their different operating principles make them suitable for very distinct types of control tasks, providing a specific example for each in an industrial setting. (Approx. 150-200 words) (total 10 marks)

DoK 4 Project Task

Teachers may decide whether this should be a group project, individual research or be presented as a lesson-based case study depending on what unautomated industries are present locally.

Total 40 (+2) marks (two extra marks are available for the recognition of a solution that shows exceptional innovation and creativity)

Project: Actuator Selection for a Local Industrial Automation Need

You are tasked with analysing a specific automation need in a local industrial context (e.g., a small manufacturing unit, an agricultural processing facility, a water pumping station). Based on this analysis, you will recommend the most suitable advanced industrial actuator(s) to serve as the final control element(s) and justify your choice.

Your project must include:

1. **Analysis of Industrial Automation Need**

   a. Describe a specific task or process within a local industrial context that could benefit from automation. Clearly explain what needs to be controlled (e.g., movement, flow, electrical power).

   b. Identify and describe the key industrial control requirements for automating this task (e.g., required force/speed, precision, type of motion/action, power available, environmental considerations, safety).

2. **Actuator Recommendation**

   - Based on your analysis of the requirements and your understanding of advanced industrial actuators (valves, hydraulic actuators, contactors, relays), recommend the most appropriate actuator(s) to be used as the final control element(s) for this specific automation need.

3. **Justification of Recommendation**

   - Provide a detailed justification for your recommended actuator(s). Explicitly explain *why* the chosen actuator(s) are suitable for meeting the identified requirements. Reference the operating principle and capabilities of the recommended actuator(s) and link them directly to the specific needs of the task and the context of the local industry. Explain why other actuator types discussed are less suitable.

4. **Documentation and Presentation**

   - Document your analysis, recommendation, and justification in a clear and well-structured report or presentation.

PLC Session 9

45 minutes	Planning & Reviewing the Learning Plan

1. Begin the session by recapping lessons from the previous year/session/week, focusing on how learning was applied.
2. Use your subject specific app to explore the content before planning. Ask the app to
   a. explain the week's concepts in a practical way that will make it easier for learners to understand
   b. identify two or three common misconceptions learners may have on the week's content and how to address them using contexts familiar to Ghanaian learners. (NTS 3a, 3g and 3m).
3. Use your subject specific app to plan your lesson. Ask the app to
   a. suggest the most effective method for teaching the week's concepts to learners with diverse backgrounds.
   b. suggest learning activities, differentiation and essential questions based on the method of teaching (NTS 3a-3q)
   c. review your prepared learning plan and address any challenging issues you may have identified in your subject area (NTS 3a-3c).

Note

1. Take note of the recap topics linked to Years 1 and 2 while preparing for your lesson.
2. Refer to the Values Learning Community Handbook and link the values for the week in your lesson to ensure that learners are applying what they learn.
3. Plan the week's key assessment with your app
   a. The key assessment for the week is ***project***. Chat with your app to plan this assessment, indicating what you and your learners will do before, during and after the project (NTS 3k-3q).

10 minutes	Content Exploration

4. As a subject group, select one problem or idea/term in the week's content that is challenging (not a routine procedural question, but one that requires conceptual understanding) and then
   a. use the app, ask it to solve the problem step by step and explain the reasoning behind each step (you may be assisted by a teacher who is strong in the content area)
   b. all teachers in the group work through a similar problem individually for 3–4 minutes

c. the group then works through the problem together step by step
d. highlight the most common errors made at each step and discuss (NTS 1c, 1e, 2c, 2e, 3e-3j).

Note

1. This part is not just about 'How do I teach this?' but how do I teach it for learners to understand?
2. If the group still shares the same content gap, use the app: ask it to solve the problem step by step and explain the reasoning behind each step.
3. Problems/ideas discussed should be sourced from the Teacher Manual content, not invented separately. This keeps the content exploration directly connected to what teachers will deliver.

30 minutes	Enactment

5. Enact a teaching activity based on your learning plan (NTS la-1b, 1e-lg, 2c, 2e, 2f, 3c-3j and 3o).
   a. Colleagues observe with specific attention to clarity of concept explanation, use of familiar Ghanaian examples and how misconceptions are handled if they arise.
   b. After the enactment, ask for structured feedback from the group
      i. was the concept explained in a way that a learner with no prior knowledge could follow?
      ii. were the examples grounded in contexts familiar to Ghanaian learners?
      iii. did any misconceptions surface and if so, how were they addressed?

5 minutes	Reflection

6. Reflect and share your views on the session. One thing that worked well, what did not work well and what you will do differently. (NTS 1a, 1b).
7. Identify a critical friend to observe and provide feedback on your lesson in relation to the PLC Session conducted (NTS 1f, 1g).
8. Remember to
   a. transfer your chats into your learning planner template.
   b. read your Teacher Manual and related Learner Material in preparation for the next session. Take note of the recap weeks, which are linked to Years 1 and 2 (NTS 3a).
   c. integrate National Values, GESI, SEL, and 21st century skills into your learning activities, assessment tasks, and rubrics/mark schemes.

Section Review

Section 4 advanced learners' understanding of sensors and actuators by focusing on their application in industrial automation. It began by analysing the operating principles of advanced industrial actuators such as valves, hydraulic systems, contactors, and relays. This provided insight into how these final control elements function in real-world automated processes. The section then moved to evaluating industrial control requirements, teaching learners to select the most appropriate actuators for specific tasks and justify their choices based on technical needs and the local industrial context.

ADDITIONAL READING

Resource	QR Code to Accessing Resource
1. Solenoid Valve Explained | Types and Application (Video)	
2. Introduction to Hydraulic Cylinders: Cylinders, Inc. (2024, August 14). What is a hydraulic cylinder? How it works, types, uses. Cylinders, Inc.(Web Publication)	
3. Understanding Electrical Contactors: RS Components. (n.d.). A complete guide to contactors. RS Components. (Web Publication)	
4. Relay vs Contactor Comparison: Wevolver. (2023, September 19). Contactor vs Relay: Understanding the differences and applications. Wevolver. (Web Publication)	
5. Overview of Industrial Actuator Types: Tameson.com. (2023, February 10). What is an actuator – Types and applications. Tameson.com. (Web Publication)	

MARKING SCHEMES FOR THE KEY ASSESSMENTS

WEEK 9

MARKING SCHEME: SHORT ANSWER QUESTIONS (TOTAL 14 MARKS)

Q1 (***2 marks***)

1. ***1 mark:*** Identifies that it is used to control the flow of fluid or gas
2. ***1 mark:*** Mentions activation via an electric current/magnetic field

Q2 (***2 marks***)

1. ***1 mark:*** States that hydraulic actuators use liquids, pneumatic use gases
2. ***1 mark:*** Recognizes the difference in force output or application environment

Q3 (***2 marks***)

1. ***2 marks:*** Correctly identifies the electromagnetic coil or solenoid

Q4 (***4 marks***)

1. ***2 marks***: Explains structural difference (e.g., contactors are built for higher current loads and often include arc suppression)
2. ***2 marks***: Describes usage in industrial/high-power switching compared to relay use in control systems

Q5 (***4 marks***)

1. ***2 marks***: Explains how fluid pressure acts on a piston to create movement
2. ***2 marks***: Elaborates on why this enables higher force transmission and is ideal for heavy-load tasks

MARKING SCHEME EXTENDED RESPONSE QUESTIONS (TOTAL 30 MARKS)

QUESTION 1 DETAILED MARKING SCHEME (10 marks each)

Scenario: Automating steam flow into a cooking vat to maintain temperature.

1. Analysis: Identify the industrial control requirement (***3 marks***)

MARKS	DESCRIPTION
3	Identifies the requirement as automatic regulation of steam flow to maintain precise temperature, linking it to process control and temperature stability.
2	Identifies steam control to maintain temperature, but lacks some detail or technical vocabulary
1	Mentions steam or temperature, but vague or partially correct.
0	Incorrect or missing answer.

2. Recommendation of Suitable Final Control Actuator (***2 marks***)

MARKS	DESCRIPTION
2	Recommends a control valve (e.g., modulating or motorised valve), which is appropriate for regulating steam flow.
1	Suggests a plausible actuator (e.g., relay, contactor) but less appropriate for fluid control.
0	No relevant actuator recommended.

3. Justification of Recommendation (***5 marks***)

MARKS	DESCRIPTION
5	Explains that valves control flow rate by varying opening size, using pressure differentials; links to precise temperature regulation, and shows deep understanding of the operating principle.
4	Mostly accurate justification, with clear reference to how valves modulate fluid flow and their role in temperature control.
3	Some explanation of the valve's function but lacks technical clarity or link to temperature control.
2	Superficial explanation; vague on how it works or why it's suitable.
1	Attempted explanation with minimal relevance.
0	No justification or irrelevant content

QUESTION 2 DETAILED MARKING SCHEME (10 marks)

Scenario: Automating power control to a large electric motor

1. **Analysis**: Identify the industrial control requirement (*3 marks*)

MARKS	DESCRIPTION
3	Identifies automated power switching for a large electric motor from a central control panel; recognises safety and centralisation needs.
2	Identifies control of motor power but with limited scope or detail.
1	Vague mention of turning motor on/off.
0	Incorrect or missing answer.

2. Recommendation of Suitable Final Control Actuator (*2 marks*)

MARKS	DESCRIPTION
2	Correctly recommends a contractor, suitable for high-power switching.
1	Suggests a general actuator (e.g., relay), partially correct.
0	No valid actuator provided.

3. Justification of Recommendation (*5 marks*)

MARKS	DESCRIPTION
5	Justifies contactor use by referencing electromagnetic operation, high current capacity, and safe electrical isolation. Explains how the contactor handles heavy loads without manual intervention.
4	Provides a mostly clear explanation with appropriate reasons and accurate technical details.
3	Adequate explanation of operating principle and suitability, with some technical errors or omissions
2	Limited understanding shown; lacks connection between features and requirements.
1	Minimal relevance or very basic response.
0	No justification or irrelevant content

QUESTION 3 DETAILED MARKING SCHEME (10 marks)

Task: Compare hydraulic actuators vs contactors in terms of application, advantages, and operating principles.

1. Comparison of Applications/Advantages (***4 marks***)

MARKS	DESCRIPTION
4	**Contrasts applications:** hydraulic actuators for heavy-duty, precise motion control, contactors for electrical switching. Discusses at least two key advantages for each.
3	Good comparison with at least one valid application and advantage for each type.
2	Limited comparison or advantages discussed for only one.
1	Mentions steam or temperature, but vague or partially correct.
0	Basic mention of application without real comparison.

2. Explanation of Operating Principles and Suitability (***4 marks***)

MARKS	DESCRIPTION
4	Explains that hydraulic actuators use pressurised fluid to create linear/rotary motion, ideal for force-intensive tasks; contactors use electromagnets to close electrical contacts, suitable for high-voltage switching. Links principles to applications.
3	Mostly accurate with both principles discussed and some link to their roles.
2	Basic explanation of one principle with little link to suitability.
1	Minimal or vague explanation.
0	No correct explanation provided.

3. Relevant Industrial Examples (***2 marks***)

MARKS	DESCRIPTION
2	Provides clear, realistic examples for both: e.g., hydraulic actuator in a robotic arm press, contactor in a motor starter circuit.
1	One example given or both examples vague.
0	No relevant example.

MARKING SCHEME: PROJECT TASK (TOTAL 40 + 2 MARKS)

1. Analysis of Industrial Automation Need: *10 Marks*

Criteria	Description	Marks
Identification of Context	Clear and relevant description of a local industrial context (e.g., small factory, farm, water station)	2
Description of Task/ Process	Accurate identification of the specific task or process to be automated	2
Definition of Control Requirement	Clearly states what is to be controlled (e.g., motion, flow, power)	2
Specification of Control Needs	Includes factors such as force/torque, precision, motion type, power type, environment, and safety	4

2. Actuator Recommendation: *8 Marks*

Criteria	Description	Marks
Selection of Actuator(s)	Relevant and appropriate actuator(s) chosen for the task	3
Link to Control Requirements	Recommendation aligns logically with the specified control needs	3
Consideration of Multiple Options	Acknowledges alternatives, even if briefly	2

3. Justification of Recommendation: *14 Marks*

Criteria	Description	Marks
Operating Principle	Clearly explains how the actuator works (e.g., electromagnetism, fluid pressure)	3
Capability Match	Demonstrates how the actuator's capabilities meet the task's requirements	4
Industry Fit	Shows understanding of the actuator's typical industrial usage and appropriateness for the local context	3
Comparison with Alternatives	Thoughtfully explains why other actuator types are less suitable	4

4. Documentation/Presentation Clarity: *8 Marks*

Criteria	Description	Marks
Structure & Clarity	Report/presentation is well-organized and easy to follow	2

Technical Language & Terminology	Accurate use of relevant technical terms (actuators, signals, final control elements, etc.)	2
Visuals & References	Includes diagrams, figures, or links (e.g., videos, QR codes) where appropriate	2
Professionalism	Overall neatness, grammar, coherence, and effort	2

OPTIONAL BONUS (UP TO +2 MARKS)

Exceptional innovation or creativity in matching an actuator to a local, real-world problem (e.g., proposing a hybrid solution, custom adaptation, or energy-saving consideration).

Grand Total: ***42 Marks* (*40 + 2 bonus marks*)**

SECTION 4: DIGITAL AND ANALOGUE SYSTEM DESIGN 3

STRAND: ROBOT DESIGN METHODOLOGIES

Sub-Strand: Digital and Analogue System Design

Learning Outcome: *Reverse engineer an existing combinational circuit and implement similar circuit or improved versions.*

Content Standard: *Demonstrate practical skills in the design and implementation of combinational digital systems that solve actual problems.*

INTRODUCTION AND SECTION SUMMARY

This lesson is strategically placed in Week 10, building directly upon the comprehensive understanding of reverse-engineering and control logic analysis that learners acquired in Weeks 5 to 8 within the context of autonomous robotic systems. The focus now transitions from complex electromechanical systems to the fundamental digital circuits that underpin such systems. Learners will apply their established analytical skills to decipher the logic of digital "black boxes" and then leverage that insight to design and implement their own combinational circuits for specific real-world problems.

The week covered by the section is

Week 10

- Observe the input/output pairs of combinational circuit black-boxes and reverse-engineer the circuit.
- Analyse a given real world scenario and build a circuit that a number of inputs to control a given number of outputs

SUMMARY OF PEDAGOGICAL EXEMPLARS

The pedagogical exemplars in this section, detail a structured instructional sequence for combinational circuit design. Instruction commences by explicitly connecting new concepts to learners' previous experiences in robotic system reverse-engineering. Guided practical sessions support systematic observation and truth table construction. The teacher manual progresses to deriving and simplifying Boolean expressions through the use of Karnaugh Maps. Learners are then guided in documenting their designs via logic diagrams. Ultimately, lessons conclude with circuit implementation and rigorous testing, which incorporates thorough troubleshooting.

Throughout these stages, strategies for differentiation and inclusivity ensure the learning content is accessible to all learners.

ASSESSMENT SUMMARY

The following table presents a sample of the key assessments used in this section.

Indicators	DOK Level	Assessment Strategy and/or Mode
Observe the input/output pairs of combinational circuit black-boxes and reverse-engineer the circuit. Analyse a given real world scenario and build a circuit that has a number of inputs to control a given number of outputs	3	Short and Extended Response Essays - (Suggested Modes: Individual Classwork, Mid-semester Assessment, End-of-semester Assessment)
		Individual Practical Task - (Suggested Modes: Individual Project Assessment)

WEEK 10

Learning Indicators

1. *Observe the input/output pairs of combinational circuit black-boxes and reverse-engineer the circuit.*
2. *Analyse a given real world scenario and build a circuit that has a number of inputs to control a given number of outputs*

FOCAL AREA 1: REVERSE ENGINEER A COMBINATIONAL CIRCUIT BLACK-BOX

Introduction

In recent studies (***Weeks 5-8***), reverse-engineering of robotic systems was explored, with focus placed on observing overall behaviour, identifying inputs from sensors and outputs to actuators, and deducing control logic, often represented through flowcharts or pseudocode. This skill of deconstructing systems proved valuable. In Week 10, these analytical skills are transferred to combinational digital circuits. A small electronic "black box" with switches (inputs) and indicator lights (outputs) is examined, where outputs depend solely on current inputs, lacking memory of past actions. Systematic observation of all input combinations is conducted to create a truth table, followed by translation into a Boolean expression to reveal hidden logic gates. This reverse-engineering capability is essential for troubleshooting electronics in automated systems, understanding proprietary components, or enhancing designs for Ghanaian applications.

Building on Foundational Concepts

This skill builds on their prior knowledge from years 1 and 2 and the earlier sections of year 3; especially Weeks 5 - 8.

Recap Weeks

Year 1	**Week 5** • Evaluate the use of logic and loop diagrams and demonstrate their use in control systems' design. **Week 6** • Draw a parallel relationship that relates the coordination of senses, brain, and moving parts in living organisms to the coordination of sensors, processors, and actuators in robots. **Week 9** • Identify the components of an electronic circuit and their functions. **Week 10** • Properly label and explain block and schematic diagram representations of electronic systems. **Week 11** • Assemble and test electronic circuits on a solderless breadboard using pre-designed schematic diagrams. **Week 17** • Define solutions to basic automated and robotic problems using algorithms, pseudocodes, and flowcharts diagrams.

Year 2	**Week 9** • Learners should explain and apply Boolean Algebra in the definition and design of digital systems. **Week 9** • Learners should apply Karnaugh Maps in the optimization of digital systems. **Week 9** • Design, build and test basic combinational circuits on Printed Circuit Boards (PCBs).

KNOWLEDGE CONTENT

The Art of Reverse-Engineering Combinational Circuits

Recall from Weeks 5 & 6 that reverse-engineering involves systematically analysing a system to understand its components, mechanisms, and, crucially, its control logic. For combinational digital circuits, this process is refined to focus on the direct, memory-less relationship between inputs and outputs.

The Truth Table: The Definitive Behavioural Map of a Combinational Circuit

As you learned when deducing the control logic of a robot from its observed actions, a truth table serves as the comprehensive blueprint for a digital circuit. It lists every possible input condition and the exact output(s) that the circuit produces for each condition.

1. **Systematic Observation:** Just as you would observe a robot's response to different sensor readings (inputs) and environmental conditions, here you will systematically apply every binary combination (0s and 1s) to the circuit's inputs.
2. **Recording Outputs:** For each input combination, you will meticulously record the resulting output value(s) (also 0s or 1s). A logical '0' typically represents a low voltage (e.g., 0 Volts), and a logical '1' represents a high voltage (e.g., 5 Volts). These can be visually indicated by the state of an LED (off for '0', on for '1').

How to Construct a Truth Table from a Physical "Black-Box" Circuit (Applying Weeks 5 & 6 Observation Skills)

Consider a scenario set in a bustling local Ghanaian company specialising in preparing fruits for beverage production. A small, unidentified circuit module, resembling a basic control unit within their automated orange sorting machine, is provided for analysis. This machine, crucial for selecting the best oranges for their popular orange drink, relies on precise automation to ensure customer satisfaction.

The circuit features three input terminals:

- **Sensor A:** Detects the ripeness of the fruit (ripened = 1 or unripened = 0).
- **Sensor B:** Detects the colour of the fruit (yellow = 1 or green = 0).
- **Sensor C:** Detects the sweetness of the fruit (sweet = 1 or sour = 0).

The single output terminal (**Z**) showcases whether a fruit is selected or not for the drink-making process. The output terminal Z is connected to an LED, which lights up when a fruit is selected. The objective is to map the circuit's function by observing how different combinations of ripeness, colour, and sweetness affect whether an orange is selected for the delicious drink. This directly impacts the final orange drink's quality and taste, ensuring it meets customer preferences.

In order to start the reverse engineering process for the 'black box' follow these points:

1. **Identify and Label:** Clearly label the input terminals (e.g., A, B, C) and output terminal(s) (e.g., Z).
2. **Prepare for Documentation:** Draw a blank truth table with columns for all inputs and outputs. For three inputs, remember there will be 23 = 8 unique input combinations. Systematically list these, often starting from 000 up to 111 as depicted in Table 10.1.

Table 10.1: Blank Truth Table with 3 inputs (A, B, C) and 1 output (Z)

A	B	C	Z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

3. **Apply Inputs Systematically:** For each row in your truth table, set the physical input connections (e.g., using switches or jumpers) to the corresponding binary values.

Hint

- *Always ensure power is disconnected before making or changing any physical connections.*

4. **Observe and Record Outputs:** Once inputs are set, connect the power supply and observe the output (e.g., an LED connected to Z). Record '0' if the LED is off, and '1' if it is on.
5. **Iterate and Complete:** Repeat this process for all 8 input combinations until the entire truth table for Z is accurately filled.

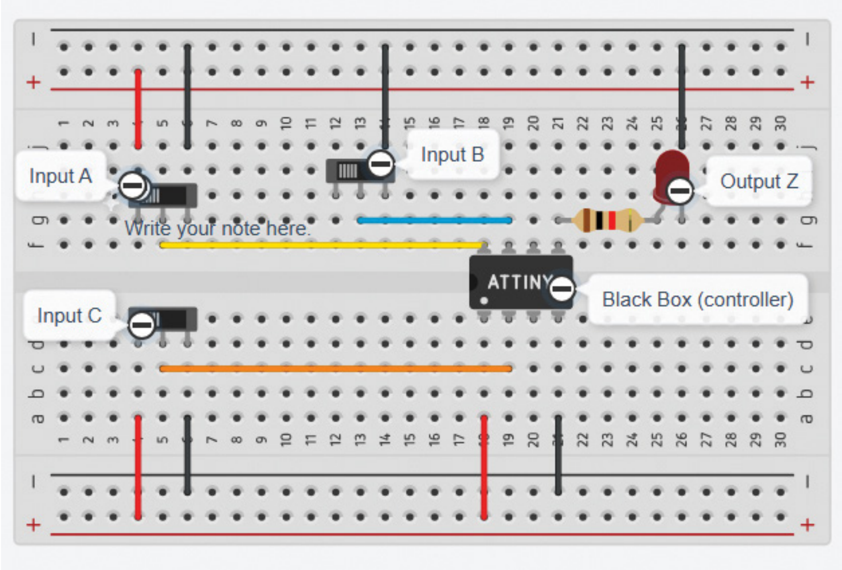


Figure 10.1: *A simplified "black box" circuit with inputs controlled by switches and an output indicated by an LED. This illustrates the systematic observation process.*

Boolean Expressions: Formalising the Inferred Control Logic (Extending from Weeks 5 & 6 Pseudocode/Flowcharts)

Once your truth table is complete, the next step in reverse-engineering is to translate this behavioural data into a mathematical language. Just as you translated a robot's decision-making into pseudocode or flowcharts, for digital circuits, we use Boolean Algebra.

1. **Sum-of-Products (SOP) Form:** Remember that this is the standard method for directly deriving a Boolean expression from a truth table.
   a. **Focus on '1's:** Identify every row in the truth table where the output is '1'. These are the conditions under which the output is active.
   b. **Create Product Terms:** For each '1' in the output, write a 'product term' (an AND combination) of the input variables.
      i. If an input is '0' in that row, use its complemented form (e.g., or A' for 'NOT A').
      ii. If an input is '1' in that row, use its uncomplemented form (e.g., A).
2. Combine with OR: Join all these individual product terms together using the OR (+) operator.

Continuing with the orange fruit sorting machine example in the previous section, imagine if the truth table, after being drawn, looks like what is present in Table 10.2. [This truth table reflects the operation of logic gates contained in the 'black box'. Selection of fruit is therefore more complicated than if circuit's goal was to detect only the "fully desirable fruit" — that is, ripened **and** yellow **and** sweet (111), a simple three input AND gate.]

Table 10.2: Truth Table for an Orange Fruit Sorting Machine with 3 inputs and 1 output

A	B	C	Z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

The rows where 'Z' is '1' are:

1. $A = 0, B = 0, C = 1 \rightarrow A^{\wedge\prime} B^{\wedge\prime} C \text{ or } \bar{A}\ \bar{B}C$
2. $A = 0, B = 1, C = 0 \rightarrow A'BC' \text{or } \bar{A}B\bar{C}$
3. $A = 1, B = 0, C = 1 \rightarrow AB'C \text{ or } A\bar{B}C$
4. $A = 1, B = 1, C = 0 \rightarrow ABC' \text{ or } AB\bar{C}$

The Boolean expression for 'Z' would be:

1. $\boldsymbol{Z} = \bar{A}\,\bar{B}C + \bar{A}B\bar{C} + A\bar{B}C + AB\bar{C}$

Logic Simplification: Optimising the Deduced Logic

Just as you learned to troubleshoot and optimise robot algorithms in Weeks 7 & 8, Boolean expressions derived directly from truth tables are often verbose and can be simplified. Simplifying means reducing the number of terms or variables, which translates directly into using fewer logic gates in a physical circuit. This leads to:

1. Lower manufacturing costs.
2. Reduced power consumption.
3. Smaller circuit size.
4. Potentially faster operation.

This simplification can be done using Karnaugh Maps which is a graphical method for simplifying Boolean expressions, which learners were introduced to in year 2. Remember that this method of simplification visually groups terms that can be logically combined.

Review of K-Map Principles

Recall how K-maps arrange truth table values in a grid using Gray code, ensuring adjacent cells differ by only one variable. This property is key for forming groups. The grouping is done using the following grouping rules:

1. Groups must contain a number of '1's that are a power of two (2, 4, 8, etc.).

2. Groups must be rectangular or square.
3. Groups can wrap around the edges of the map (top-to-bottom, left-to-right).
4. Always form the largest possible groups first.
5. Ensure every ‘1’ is included in at least one group, using the minimum number of groups.

For each group, identify the variables that remain constant (either ‘0’ or ‘1’) across all cells in that group. Variables that change within the group are eliminated. Combine the constant variables (complemented if ‘0’, uncomplemented if ‘1’) to form a simplified product term for that group. Finally, sum (OR) all these simplified product terms.

So, for the example provided earlier, the orange fruit sorting machine, the simplified Boolean expression can be generated using a K-Map as presented in Figure 10.2.

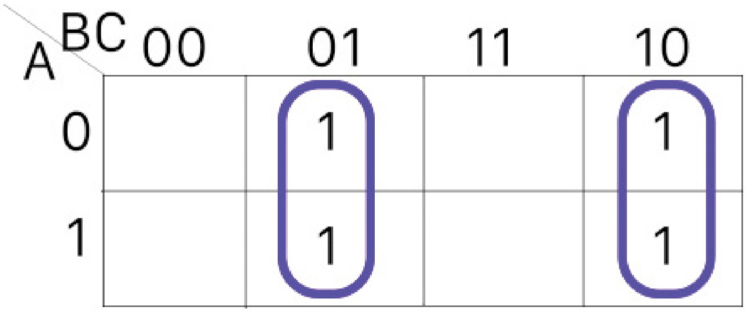


Figure 10.2: *K-Map of the Orange Fruit Sorting Machine Black-box*

Thus, the above K-Map is used to simply the initial Boolean expression above from
$\bar{A}\,\bar{B}\boldsymbol{C} + \bar{A}\,\boldsymbol{B}\bar{C} + \boldsymbol{A}\bar{B}\boldsymbol{C} + \boldsymbol{AB}\bar{C}$ **to** $\boldsymbol{Z} = \bar{B}\boldsymbol{C} + \boldsymbol{B}\bar{C}$

This simplified expression is equivalent to the XOR functionality where $\boldsymbol{Z} = \boldsymbol{B} \oplus \boldsymbol{C}$

Circuit Implementation: From Logic to Hardware (Putting the Design into Action)

The final step is to build the circuit based on the simplified design. This can be done through the following steps:

1. **Logic Gate Diagram:** Using the simplified Boolean expression, draw a schematic using standard logic gate symbols (AND, OR, NOT). This is your blueprint for the physical circuit.

   For the example presented earlier of the Orange Fruit Sorting Machine, whose simplified expression was derived as $\boldsymbol{Z} = \bar{B}\,\boldsymbol{C} + \boldsymbol{B}\bar{C}$ **or** $\boldsymbol{Z} = \boldsymbol{B} \oplus \boldsymbol{C}$, its logic gate diagram would look as depicted in ***Figure 10.3.***

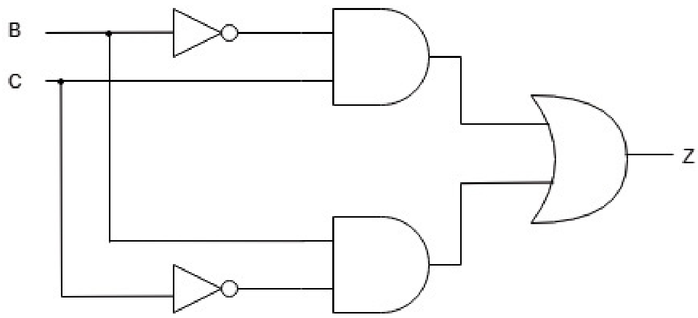


Figure 10.3: *Logic Gate Diagram for* $Z = \bar{B}C + B\bar{C}$

Or even more simply as presented in ***Figure 10.4***.

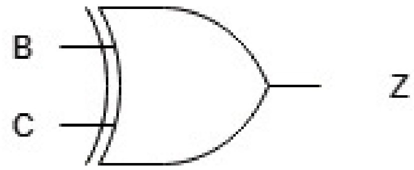


Figure 10.4: *Logic Gate Diagram for* $Z = B \oplus C$

2. **Component Selection:** Choose appropriate Integrated Circuits (ICs) from a digital logic family (e.g., 74LS series) that contain the necessary gates (AND, OR, NOT).
3. **Physical Wiring/Simulation:** Next is to wire the selected components together either physically or using a software simulation tool.
   a. **Physical Prototype:** Using a breadboard, carefully connect the ICs, inputs (switches), and outputs (LEDs) according to your logic diagram. Pay attention to IC pinouts (e.g., power, ground, inputs, outputs).
   b. **Software Simulation:** Use a digital circuit simulator to build and test your circuit virtually. This is a safe and efficient way to verify your design before physical construction.
4. **Testing and Verification:** More importantly, apply every input combination from your original truth table to your built/simulated circuit. Verify that the actual outputs match your desired outputs from the truth table. If not, systematically troubleshoot your wiring, or revisit your Boolean expression and K-map simplification. This iterative troubleshooting is a key skill from Weeks 7 & 8.

Real-World Applications

The knowledge gained above can be applied in various real-world scenarios such as the following:

1. **Automated Gate Control:** Designing a circuit for a factory gate that opens only if certain security conditions (e.g., valid ID card AND vehicle sensor activated) are met.
2. **Irrigation Control:** A circuit that activates a pump only if soil moisture is low AND it is between sunrise and sunset.

3. **Safety Interlocks:** Ensuring a machine in a textile factory operates only when all safety covers are closed AND the emergency stop is not pressed.
4. **Traffic Light Intersection Logic:** Simple combinational logic can manage light changes based on vehicle presence sensors and timers.

Learning Tasks

The following tasks build upon learners' existing reverse-engineering capabilities, applying them specifically to digital combinational circuits and then extending to their mathematical representation and simplification. Depending on the available time or resources, administer one or more of the following learning tasks to help learners reinforce understanding and acquire new knowledge or skills.

- You will be provided with an unidentified physical "black box" combinational circuit (e.g., a small breadboard circuit with 3 input switches and 1 output LED, perhaps simulating a simple control panel from a Ghanaian factory). Systematically apply all possible input combinations to the circuit. Observe and record the corresponding output(s) in a meticulously constructed truth table.

Hint

- *This task directly leverages the "External Observation and Behavioural Analysis" skills you developed in Weeks 5 & 6 when analysing robotic systems.*

1. Using the truth table derived from Task 1, you will now formalise the circuit's logic by carrying out the following activities:
   a. Derive the Sum-of-Products (SOP) Boolean expression for each output of the "black-box" circuit.
   b. Apply Karnaugh Maps to simplify each Boolean expression to its most minimal form. Show all groupings clearly on the K-map.
   c. Based on the simplified expression, describe in words the fundamental logical function that the "black-box" performs (e.g., "This circuit acts as an OR gate," or "This is a 3-input majority vote circuit").

Hint

- *This task extends your "Inferring Control Logic" skill from Weeks 5 & 6 (translating pseudocode to Boolean expressions) and applies your "Iterative Fault Detection and Correction" skills from Weeks 7 & 8 (through optimisation).*

2. Based on your simplified Boolean expression from Task 2, draw the logic gate diagram for the simplified circuit using standard logic gate symbols (AND, OR, NOT, etc.).
3. Based on your drawn logic gate diagram from Task 3, implement the circuit using one of the following methods:

a. **Option A:** Software Simulation: Use a digital circuit simulation software to build your logic gate diagram.

b. **Option B:** Physical Prototype (if resources permit): Using a breadboard and standard logic IC chips (e.g., 74LS series gates), physically build your circuit.

4. Test and verify the circuit by systematically applying every input combination from your original truth table to your implemented circuit (simulated or physical). Observe and document the resulting output(s). If the outputs do not match the truth table, identify the source of the error (e.g., wiring, K-map mistake, incorrect Boolean logic) and correct it.

PEDAGOGICAL EXEMPLARS

These exemplars provide a guideline for administering the lesson, ensuring that it seamlessly integrates with prior learning while progressively introducing new depths of understanding and practical application in digital logic.

1. **Activating Prior Knowledge: From Robotics to Digital**

   To explicitly link learners' existing knowledge of reverse-engineering robots to the concept of reverse-engineering combinational circuits. Begin with a brief, engaging recap of Weeks 5-8. Ask learners to describe in their own words:

   a. "What is reverse engineering in robotics?" (Expect answers about understanding a robot's internal workings from its external behaviour)

   b. "How did you figure out a robot's decision-making process?" (Expect answers about observing inputs, outputs, deducing logic, flowcharts/pseudocode).

   Then transition by introducing the concept that a digital circuit is a simplified "black box" where inputs (like sensor signals) directly determine outputs (like actuator commands), but without the complexity of memory or physical movement. Emphasise that the same systematic observation and logical deduction skills are directly transferable. Show a simple image of a binary switch and an LED as inputs/outputs for a digital circuit.

To ensure differentiation and inclusivity:

- Provide large-print handouts and audio descriptions for visually impaired learners.
- Offer pre-written prompts for learners with cognitive challenges to aid recall.
- Use tactile models of switches and LEDs for hands-on exploration by visually impaired learners.
- Allow verbal responses or peer support for learners with writing difficulties.
- Present the image with captions and narrations for inclusivity across abilities.
- Ask more complex questions requiring learners to explain how different components might work together in a simple system (e.g., how sensors and actuators function collaboratively for navigation).

- Verify that visual aids are clear and easily seen. Provide key terms in written format for learners with hearing impairments. Offer various methods for learners to participate, including verbal responses and written notes.

2. **Step 2: Guided Practical Observation and Truth Table Construction (Learning Task 1)**

   Guide learners through the hands-on process of systematically observing a physical "black box" digital circuit and accurately documenting its behaviour in a truth table. Divide learners into small groups. Provide each group with a pre-assembled "black box" combinational circuit (e.g., a simple circuit on a breadboard with 2-3 input switches and 1-2 output LEDs). Thoroughly review electrical safety procedures. Insist that power is disconnected during wiring and only applied when instructed.

   Thoroughly review electrical safety procedures. Insist that power is disconnected during wiring and only applied when instructed. Learners then proceed to complete Task 1. Circulate among groups, providing prompts ("What is the state of input B for this row?", "What is the output LED doing now?") and checking for methodical observation.

To ensure differentiation and inclusivity

- Provide circuits with only 2 inputs for some groups or offer partially pre-filled truth table templates.
- Provide circuits with 3-4 inputs or multiple outputs for others, encouraging them to deduce patterns before completing all rows.
- Ensure circuit components are clearly visible and accessible. For learners with fine motor skill challenges, pre-wired connections or larger switches can be beneficial. Encourage verbalisation of observations within groups for auditory learners.

3. **Step 3: Boolean Expressions and K-Maps (Learning Task 2)**

   Once truth tables from Task 1 are verified, transition to Task 2. Briefly review (or model if necessary) how to derive the Sum-of-Products (SOP) Boolean expression from a truth table, focusing on the '1's in the output column. Connect this back to how they would write logical conditions in pseudocode.

   Revisit the K-map concepts covered in the prior digital logic lesson. Emphasise the benefits of simplification (fewer gates, efficiency). Model transferring truth table '1's to a K-map and clearly demonstrate the rules for optimal grouping (largest groups, powers of two, wrap-around). Show how each group simplifies to a product term. Learners apply these steps to their truth tables from Task 1. Circulate, checking their derivations and K-map groupings.

To ensure differentiation:

- Provide K-map templates with grey code labels already filled. Offer pre-drawn (but unsimplified) logic gate diagrams that students then simplify.
- Ask advanced learners to attempt to derive the Product-of-Sums (POS) expression as well, or to identify which standard logic gate (if any) their simplified circuit resembles.

- Use large, clear diagrams on the board or projector to illustrate K-map groupings. Encourage peer-teaching within groups.

4. **Step 4: Documenting the Design using Logic Diagrams (Task 3)**

   Guide learners to translate their final, simplified Boolean expressions into standard logic gate diagrams. Model drawing a simple logic diagram from a simplified expression (e.g., $Y=A \cdot B+C'$). Emphasise clear connections, standard symbols, and proper labelling of inputs and outputs.

To ensure inclusivity:

- Provide a "Logic Gate Symbol Reference Sheet."
- Ask advanced learners to consider how their simplified circuit compares to the number of gates required if they had built the unsimplified version.
- Ensure clear, large examples of gate symbols are available. Encourage collaborative drawing and peer-checking of diagrams.

5. **Step 5: Facilitating Circuit Implementation and Verification (Task 4)**

   Guide learners in building (physically or virtually) and thoroughly testing their designed circuits. Based on resources, guide learners to either:

   a. Option A **(Simulation):** Provide access to computers with digital circuit simulation software (e.g., Logisim, online simulators). Provide a quick tutorial on using the software.

   b. **Option B (Physical):** Provide breadboards, common 74LS series ICs (e.g., 74LS08, 74LS32, 74LS04), wires, switches, and LEDs. Provide clear pin-out diagrams for the ICs. Conduct a thorough safety briefing on handling electronics. Model how to place ICs on the breadboard and make initial power/ground connections.

   After building or simulating their design, instruct learners to systematically test their circuit using every input combination from their original truth table. Emphasise that this systematic testing is directly from Weeks 7 & 8 (test plan creation). If the circuit does not work, guide them through a structured troubleshooting process (check truth table, check Boolean expression, check K-map, check logic diagram, check wiring/simulation connections).

To ensure inclusivity:

- For physical building, provide pre-populated breadboards with power/ground connections. For simulation, provide pre-saved basic gate components. Work closely with struggling groups.
- Encourage advanced learners to compare the physical implementation challenges (e.g., wiring complexity, component availability) with simulation benefits. Ask them to design a small test plan for their circuit.
- Ensure clear, large print schematics and pin-out diagrams. For physical tasks, use workbenches at appropriate heights. Encourage pairing where one person can focus on reading and the other on physical manipulation.

KEY ASSESSMENT LEVEL 3: SHORT AND EXTENDED RESPONSE ESSAYS (10 MARKS)

Instructions: Answer the following questions with clear explanations and justifications.

1. A small, community-run cassava processing plant in Eastern Region, Ghana, is experiencing issues with its automated chipper. You suspect a combinational control circuit might be faulty. Describe a systematic, step-by-step approach you would take to reverse-engineer this specific circuit, assuming you can access its input and output terminals. Justify why each step is critical to successfully deduce the circuit's intended logical function, identify its original design, and diagnose any potential faults by comparing its observed behaviour to its expected operation. (*8 marks*)
2. Explain, with reference to the production of a new electronic product for the Ghanaian market (e.g., a solar lamp controller), why the simplification of Boolean expressions using Karnaugh Maps is considered a critical engineering practice rather than merely an academic exercise. Provide at least three distinct benefits that arise from this simplification. (*6 marks*)

Individual Practical Task/Project (DoK 3)

Project Title: ***The "Akwaaba" Welcome Circuit***

Scenario: A new guesthouse in Aburi, Eastern Region, wants an automated "Akwaaba" (Welcome) sign light for their entrance. This light should activate based on a combination of factors: if a guest is detected at the entrance during evening hours, or if a staff member manually activates it for special guests. The system also needs a "Maintenance Mode" which temporarily deactivates the welcome light to prevent false triggers during cleaning or repair.

The inputs are as follows:

- **G** (Guest Presence Sensor): '1' if guest detected, '0' otherwise.
- **E** (Evening Hours Sensor): '1' if it is evening (e.g., 6 PM - 10 PM), '0' otherwise.
- **M** (Manual Activation Switch): '1' if staff manually activates, '0' otherwise.
- **T** (Maintenance Mode Switch): '1' if in maintenance mode, '0' otherwise.

The output is as follows:

1. **W** (Welcome Light): '1' if the "Akwaaba" light should be ON, '0' if OFF.

Rules for Welcome Light (W):

a. The Welcome Light (W) should be ON if a guest is present (G=1) AND it is evening hours (E=1).

b. OR, the Welcome Light (W) should be ON if the manual activation switch is pressed (M=1).

c. Critical Condition: If the Maintenance Mode Switch (T) is ON, the Welcome Light (W) must always be OFF, regardless of any other input conditions.

Task

1. Create a complete truth table for the Welcome Light (W) output, including all 16 possible input combinations for G, E, M, and T. Populate the W column based on the specified rules. Explicitly note how the "Maintenance Mode" rule (T) overrides all other conditions. (10 Marks)
2. Derive the unsimplified Sum-of-Products (SOP) Boolean expression for the Welcome Light (W) output from your truth table. (5 marks)
3. Draw a Karnaugh Map for the W output and use it to simplify the Boolean expression to its most minimal form. Clearly show all groupings on the K-map. (10 marks)
4. Draw the logic gate diagram for the simplified circuit using standard logic gate symbols. (5 marks)
5. Implement your simplified circuit using either software simulation or a physical breadboard prototype. (10 marks)
   a. For Software Simulation: Provide a clear screenshot of your functional circuit in the simulator.
   b. For Physical Prototype: Provide a clear photograph of your built circuit on the breadboard.
6. Develop a test plan by listing at least five critical input combinations that rigorously test all aspects of the rules, including the override condition. Execute your test plan on your implemented circuit (simulation or physical). Document the observed outputs for each test case. (10 marks)
7. If any discrepancies are observed between your expected and actual outputs, document the troubleshooting steps taken and how the issue was resolved. If there are no issues, state that the circuit functioned as expected. (optional)

(50 marks)

Conclusion

This week, the methodical process of reverse-engineering combinational circuits was examined. Learners were guided through systematically observing circuit inputs and outputs to construct definitive truth tables, thereby mapping a device's behavioural logic. This raw data was then formalised using Boolean Algebra, specifically the Sum-of-Products form. Significant emphasis was placed on optimising these expressions through Karnaugh Maps, a critical step for achieving cost-effective and efficient designs. The practical application of this knowledge culminated in drawing logic gate diagrams and implementing circuits either physically or via simulation. Rigorous testing and verification procedures were underscored as essential for ensuring functionality and identifying faults, preparing learners for authentic engineering challenges.

PLC Session 10

45 minutes	Planning & Reviewing the Learning Plan

1. Begin the session by recapping lessons from the previous year/session/week, focusing on how learning was applied.
2. Use your subject specific app to explore the content before planning. Ask the app to
   a. explain the week's concepts in a practical way that will make it easier for learners to understand
   b. identify two or three common misconceptions learners may have on the week's content and how to address them using contexts familiar to Ghanaian learners. (NTS 3a, 3g and 3m).
3. Use your subject specific app to plan your lesson. Ask the app to
   a. suggest the most effective method for teaching the week's concepts to learners with diverse backgrounds.
   b. suggest learning activities, differentiation and essential questions based on the method of teaching (NTS 3a-3q)
   c. review your prepared learning plan and address any challenging issues you may have identified in your subject area (NTS 3a-3c).

Note

1. *Take note of the recap topics linked to Years 1 and 2 while preparing for your lesson.*
2. *Refer to the Values Learning Community Handbook and link the values for the week in your lesson to ensure that learners are applying what they learn.*

4. Plan the week's key assessment with your app
   - The key assessment for the week is ***short and extended response essays***. Chat with your app to plan this assessment, indicating what you and your learners will do before, during and after the assessment (NTS 3k-3q).

10 minutes	Content Exploration

5. As a subject group, select one problem or idea/term in the week's content that is challenging (not a routine procedural question, but one that requires conceptual understanding) and then
   a. use the app, ask it to solve the problem step by step and explain the reasoning behind each step (you may be assisted by a teacher who is strong in the content area)
   b. all teachers in the group work through a similar problem individually for 3–4 minutes
   c. the group then works through the problem together step by step

d. highlight the most common errors made at each step and discuss (NTS 1c, 1e, 2c, 2e, 3e-3j).

Note

1. This part is not just about 'How do I teach this?' but how do I teach it for learners to understand?
2. If the group still shares the same content gap, use the app: ask it to solve the problem step by step and explain the reasoning behind each step.
3. Problems/ideas discussed should be sourced from the Teacher Manual content, not invented separately. This keeps the content exploration directly connected to what teachers will deliver.

30 minutes	Enactment

6. Enact a teaching activity based on your learning plan (NTS la-1b, 1e-lg, 2c, 2e, 2f, 3c-3j and 3o).
   a. Colleagues observe with specific attention to clarity of concept explanation, use of familiar Ghanaian examples and how misconceptions are handled if they arise.
   b. After the enactment, ask for structured feedback from the group
      i. was the concept explained in a way that a learner with no prior knowledge could follow?
      ii. were the examples grounded in contexts familiar to Ghanaian learners?
      iii. did any misconceptions surface and if so, how were they addressed?

5 minutes	Reflection

7. Reflect and share your views on the session. One thing that worked well, what did not work well and what you will do differently. (NTS 1a, 1b).
8. Identify a critical friend to observe and provide feedback on your lesson in relation to the PLC Session conducted (NTS 1f, 1g).
9. Remember to
   a. transfer your chats into your learning planner template.
   b. read your Teacher Manual and related Learner Material in preparation for the next session. Take note of the recap weeks, which are linked to Years 1 and 2 (NTS 3a).
   c. integrate National Values, GESI, SEL, and 21st century skills into your learning activities, assessment tasks, and rubrics/mark schemes.

Section Review

This section transitioned reverse-engineering principles from robotic systems to combinational digital circuits. Learners systematically observed black-box behaviour to create truth tables, formalising logic with Boolean expressions. Optimisation using Karnaugh Maps was emphasised for efficient design. Practical application involved designing and implementing circuits via simulation or physical prototypes. Rigorous testing ensured functionality and addressed faults, equipping learners for real-world engineering challenges in automated systems, including Ghanaian contexts.

ADDITIONAL READING

Resource	QR Code to Accessing Resource
1. Logic Gates and Truth Tables – Step-by-Step Lesson from the Classroom (Video Resource).	
2. Boolean Algebra and Logic Gates (Web Publication)	
3. Karnaugh Maps and Their Use in Boolean Algebra Simplification (Web Publication)	
4. DigiSim – Digital Logic Simulator (Online Simulator)	

MARKING SCHEMES FOR THE KEY ASSESSMENTS

WEEK 10

Marking Scheme - Short and Extended Response Essays

Question 1

1. **Step 1:** Identify and Isolate: Disconnect power, identify input/output terminals (e.g., from power source, start button, safety sensors to motor control). ***(1 mark)***
    - **Justification**: Essential for safety and to prevent further damage, and to clearly define boundaries of the "black box". ***(1 mark)***
2. **Step 2:** Construct Blank Truth Table: Draw a truth table with columns for all identified inputs and outputs, listing all possible input combinations (2n rows). ***(1 mark)***
    - **Justification**: Provides a systematic framework for recording observed behaviour, ensuring no input combination is missed. ***(1 mark)***
3. **Step 3:** Systematic Input Application and Output Recording: Apply each input combination to the circuit (using switches/jumpers), power on, observe the output (e.g., LED state), and record in the truth table. ***(1 mark)***
    - **Justification**: This is the core 'observation' phase, directly mapping external behaviour to internal logic. ***(1 mark)***
4. **Step 4:** Derive Boolean Expression: From the completed truth table, derive the unsimplified Sum-of-Products (SOP) Boolean expression for each output. ***(1 mark)***
    - **Justification**: Translates observed behaviour into a formal, mathematical representation. ***(1 mark)***
5. **Step 5:** Simplify Boolean Expression (K-Map): Use Karnaugh Maps to simplify the derived Boolean expression(s). ***(1 mark)***
    - **Justification**: Leads to the most efficient circuit, potentially revealing the original, simplified design or an optimal alternative, aiding fault diagnosis. ***(1 mark)***
6. **Step 6:** Draw Logic Diagram and Compare: Draw the logic gate diagram from the simplified expression and compare it to expected or known designs. ***(1 mark)***
    - **Justification**: Provides a visual representation for understanding and comparison, critical for identifying discrepancies and locating faults within the original chipper's circuit. ***(1 mark)***

(Award marks for clear steps and comprehensive justifications. Max 8 marks for 4 well-explained steps with justification, or 6 steps with slightly less detail but still accurate.)

Question 2

1. **Explanation:** Simplification reduces the number of terms or variables in a Boolean expression, which directly correlates to reducing the number of logic gates required to build the circuit. (2 marks for clear explanation)
2. **Three Distinct Benefits (1 mark each for stating, *1 mark* for justification/explanation):**
   a. **Reduced Manufacturing Costs:** Fewer gates mean less raw material and fewer components to purchase, making the final product more affordable for the Ghanaian market.
   b. **Lower Power Consumption:** Each gate consumes power. Fewer gates result in less electricity usage, which is beneficial for battery-powered devices like solar lamps or reducing operational costs for larger systems.
   c. **Smaller Physical Size/Footprint:** Circuits with fewer gates are physically smaller, allowing for more compact product designs, which can be advantageous for integration into small devices or where space is limited.
   d. **Improved Reliability/Durability:** Fewer components mean fewer points of failure, potentially leading to a more reliable product with a longer lifespan, especially important in challenging environments.
   d. **Faster Operation/Reduced Propagation Delay:** Signals pass through fewer gates, reducing the overall time it takes for inputs to propagate to outputs, leading to faster circuit response.

(***Award up to 2 marks for each benefit if clearly stated and explained/justified in the context of the solar lamp controller***.)

Rubric – Individual Practical Task/Project

Criteria	Exemplary	Proficient	Developing	Needs Improvement
1. **Truth Table & Problem Analysis (10 Marks)**	Truth table is perfectly accurate and complete (all 16 rows). All rules, especially the override, are correctly reflected. (10)	Truth table is mostly accurate with minor errors. Rules largely correctly applied. (7–9)	Truth table has significant errors or is incomplete. Key rules, especially override, are incorrectly applied. (5–6)	Truth table is substantially incorrect or missing. Rules are misunderstood. (0–4)
2. **Boolean Derivation & Simplification (15 Marks)**	Unsimplified SOP correct. K-map flawlessly drawn with optimal groupings. Simplified expression accurate and minimal. (15)	Unsimplified SOP mostly correct. K-map largely accurate with minor grouping errors. Simplified expression mostly correct. (10–14)	Unsimplified SOP has errors. K-map contains significant errors or is incomplete. Simplified expression incorrect. (7–9)	Unsuccessful in deriving Boolean expression or K-map. Significant errors in simplification. (0–6)

3. Logic Diagram & Implementation (15 Marks)	Logic diagram is correct, clear, and uses standard symbols. Implementation (simulated/ physical) is fully functional and matches diagram. (15)	Logic diagram is mostly correct. Implementation is functional with minor deviations or matches diagram mostly. (10–14)	Logic diagram has errors or is unclear. Implementation is partially functional or contains significant errors. (7–9)	Logic diagram is incorrect or missing. Implementation is non-functional or not attempted. (0–6)
4. Testing & Verification (10 Marks)	Test plan is comprehensive and effective. Testing results are accurately documented. (10)	Test plan is adequate. Testing results are mostly accurate. (7–9)	Test plan is insufficient. Testing results are inaccurate or poorly documented. (5–6)	No test plans. No testing performed or results not documented. (0–4)

SECTION 5: ROBOT PROGRAMMING

STRAND: ROBOT CONSTRUCTION & PROGRAMMING

Sub-Strand: Robot Construction

Learning Outcome: *Create robots using fabricated robotic materials or local materials to implement basic mechanics.*

Content Standard: *Final Year Project: Demonstrate ability to create fully functional prototypes of robotic systems that address an identified global issue.*

INTRODUCTION AND SECTION SUMMARY

Building upon foundational knowledge of robot construction and programming from previous years, this section equips learners with the essential programming skills to develop sophisticated robotic prototypes. For example, this section develops the programming and control approach for entire, integrated robotic systems, systematically linking the mechanical actuators, sensory feedback mechanisms, and complex decision-making logic studied in previous years. Moving beyond visual, block-based environments, it transitions learners to text-based Python to program these subsystems to function as a cohesive, intelligent, and autonomous machineLearners will master Python's syntax, variables, data types, and logical structures, building the capacity to write precise instructions for robot behaviours. This programming proficiency is crucial for designing and implementing functional robotic solutions that address identified global challenges, enabling complex operations and intelligent responses necessary for real-world applications.

The weeks covered by the section are:

- **Weeks 11 – 16:** Create robots using robotic kits and/or local materials to implement basic mechanics for actuations that make use of a selection from the following list:

GEARS

- Transmitting Rotation with rubber bands or chains
- Transmitting Rotation over long distances
- Off-centre axes of rotation
- Changeover mechanism using rotational Direction
- Universal Joints

VEHICLES

- Suspended Wheels Steering Mechanisms

MOVING WITHOUT TYRES

- Moving through vibrations

ARMS, WINGS & OTHERS

- Using attachments to change motion
- Changing the angle of rotation freely

SUMMARY OF PEDAGOGICAL EXEMPLARS

In summary the pedagogical exemplars adopt a differentiated and inclusive instructional approach, ensuring all learners engage meaningfully with technical content. Lessons begin by activating prior knowledge, followed by guided, hands-on tasks that promote conceptual understanding through experiential learning. Facilitation includes structured brainstorming, collaborative analysis, and scaffolded use of tools such as flowcharts and pseudocode, enabling learners to construct knowledge through active participation.

Throughout, talk for learning is embedded via peer critique, dialogic reflection, and teacher-led questioning, fostering metacognitive awareness and deeper reasoning. Learners are encouraged to verbalise their thinking, explain their design choices, and engage in collaborative problem-solving. Differentiation and inclusivity are sustained across all phases, empowering learners to apply and communicate technical concepts in ways that reflect their individual strengths and learning preferences.

ASSESSMENT SUMMARY

The following table presents a sample of the key assessments used in this section.

Indicator	DOK Level	Assessment Strategy and/or Mode
Create robots using robotic kits and/or local materials to implement basic mechanics for actuations that make use of the following: **GEARS** • Transmitting Rotation with rubber bands or chains • Transmitting Rotation over long distances • Off-centre axes of rotation • Changeover mechanism using rotational direction • Universal Joints **VEHICLES** • Suspended Wheels • Steering Mechanisms **MOVING WITHOUT TYRES** • Moving through vibrations **ARMS, WINGS & OTHERS** • Using attachments to change motion • Changing the angle of rotation freely	4	Short Answer Questions and Extended Response Essays – (***Suggested Modes:*** Individual Classwork/Homework, Case Study)
		Individual Project– (Suggested ***Modes***: Individual Classwork/ Homework, Case Study)

WEEKS 11 – 16

Learning Indicator

Create robots using robotic kits and/or local materials to implement basic mechanics for actuations that make use of the following:

1. ***GEARS***
   - *Transmitting Rotation with rubber bands or chains*
   - *Transmitting Rotation over long distances*
   - *Off-centre axes of rotation*
   - *Changeover mechanism using rotational Direction*
   - *Universal Joints*
2. ***VEHICLES***
   - *Suspended Wheels*
   - *Steering Mechanisms*
3. ***MOVING WITHOUT TIRES***
   - *Moving through vibrations*
4. ***ARMS, WINGS & OTHERS***
   - *Using attachments to change motion*
   - *Changing the angle of rotation freely*

FOCAL AREA 1: UNDERSTANDING PROGRAMMING PARADIGMS IN ROBOTICS

This lesson marks a significant advancement in learners' programming journey in robotics. Building upon the visual, block-based programming experiences from Year 1, this session introduces Python, a powerful and versatile text-based programming language. Learners will explore the shift from snapping code blocks together to writing precise lines of text, understanding how this transition offers greater flexibility, control, and scalability for developing sophisticated robotic systems. The focus is on establishing a strong foundation in Python's core elements: syntax, variables, data types, operators, and data structures like lists, all within the context of their practical application in robotics.

Building on Foundational Concepts

This section signifies a pivotal advancement from visual block-based programming, which learners engaged with in Year One, towards the precision and versatility of Python. Learners transition from snapping predefined blocks to writing textual code, building directly on their understanding of fundamental programming concepts such as sequencing, variables, and logical flow. The essence of this shift lies in empowering learners to translate algorithms and control system designs, previously represented in flowcharts and pseudocode, into functional, scalable robot programs. This deepens their understanding of how precise instructions control complex robotic behaviours.

Recap Weeks

Year 1	**Week 5** • Evaluate the use of logic and loop diagrams and demonstrate their use in control systems' design. **Week 13** • Explore features of selected modelling, programming and simulation tools useful for the design of robots. **Week 17** • Determine the Inputs, Processes and Outputs required to solve a particular problem. • Define solutions to basic automated and robotic problems using algorithms, pseudocodes, and flowcharts diagrams **Weeks 21 – 23** • Create computer programs from pre-designed flowcharts that have single-decision conditions • Create computer programs from pre-designed flowcharts that have nested decision conditions. **Weeks 24** • Create computer programs from pre-designed flowcharts that have a controlled feedback loop with loop-interrupts
Year 2	**Week 17** • Learners should define solutions for control and feedback systems using algorithms, pseudocodes, and flowcharts diagrams

KNOWLEDGE CONTENT

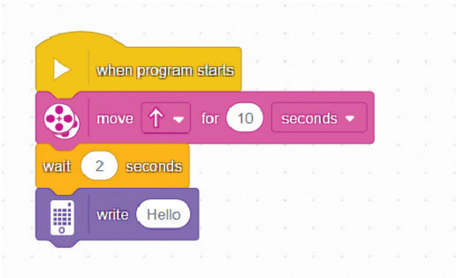
In Year One, learners were introduced to robotics through a visual block-based environment using the LEGO SPIKE™ Prime App. This visual interface enabled beginners to build logical sequences using snap-together code blocks. These blocks represent various control structures and commands, making programming more accessible by reducing concerns about syntax and focusing on program flow.

As learners advance, transitioning from block-based to text-based programming becomes essential. Text-based programming, such as with Python, involves writing code using a formal language composed of characters, symbols, and keywords. This method provides greater flexibility, precision, and power in defining complex operations. Unlike blocks that you snap together visually, Python involves typing words and symbols in a specific format (called syntax) to give instructions to a robot or computer. At first, it may feel unfamiliar, but the skills you're learning now open doors to more powerful and professional tools in robotics.

Comparison of Block Structure and Python Syntax

The primary distinction between block-based and text-based programming lies in their representation and syntax. Block-based programs are visual and rely on predefined "shapes" to ensure correct command sequencing. For example, a "move motor" block visually connects to a "wait" block. In contrast, Python syntax requires precise textual representation. Commands are written as lines of code, and the order of execution is determined by the sequence of these lines and Python's indentation rules.

Consider a simple sequence: moving a motor, waiting for a duration, and then stopping and displaying a specific text. Fig. 5.1 compares this in a block-based environment vs a text-based environment.




```
robot.motor_A.move_forward_for_seconds(10.0) # Assuming motor A runs for 10 seconds
time.sleep(2.0) # Waits for 2 seconds
print("Hello") # Prints "Hello" to the console or display
```

Figure 5.1: *Coding Sequence in a Block-Based environment vs a text-based environment*

In a block-based environment, this might involve three distinct blocks stacked vertically. In Python, this sequence would be represented by specific function calls, each on its own line as shown in ***Fig. 5.1***.

The transition to text-based Python programming is a crucial step for developing sophisticated and scalable robotic control systems that offer real practicability in today's industries. While block-based environments excel at introductory concepts, their limitations become apparent with increasing program complexity. The following are some essential reasons why it is necessary to switch to text-based coding as we develop more sophisticated programs:

1. **Scalability and Complexity Management**: You can build more complex robots and reuse parts of your code using tools called functions and modules (which we will learn later).
2. **Enhanced Control and Customisation**: Python lets us write custom behaviours, respond to sensors more precisely, and debug errors using better tools.
3. **Integration with Advanced Libraries and Tools**: Python is widely used in industries like space robotics, automation, medical robots, and more. It prepares you for professional robotic platforms.
4. **Troubleshooting and Debugging Capabilities**: Python offers detailed debugging tools that help identify and fix errors more precisely. Programmers can inspect variables, step through code, and isolate issues with greater accuracy than is possible in block environments. This makes Python more effective for resolving issues in complex robotic systems.

Switching from drag-and-drop blocks to typing Python code can be challenging at first. But there are tools designed to make this change easier for learners like you. Here are some free tools you can explore:

1. **EduBlock:** EduBlock lets you learn Python by using blocks first. As you drag and drop blocks to build a program, EduBlock shows you the real Python code being generated. This helps you see the connection between block commands and text-based code in real time. It's a great way to build confidence before writing full Python programs on your own.
2. **Tinker Coders:** Tinker Coders is an online platform that helps students learn Python and robotics in a fun and interactive way. It uses step-by-step lessons and block-to-text activities that guide you from beginner to more advanced skills. You can even simulate robots and test your code in a virtual environment—great for practicing when you don't have a physical robot.
3. **Blockly:** Blockly is a visual tool developed by Google that works like LEGO SPIKE blocks. It allows you to create programs by dragging blocks, and then it translates those blocks into Python, JavaScript, or other code. This helps you get used to reading and understanding text code while still using the blocks you're familiar with.

Link	QR CODE
Getting started with Python coding using Edublocks https://www.youtube.com/watch?v=FEKEsEnUnnQ	

For educators and students, weighing the benefits of such software against budget constraints is essential. Open-source platforms like MU Editor or Thonny (with MicroPython support) can serve as cost-effective supplements for practicing text-based robotics programming.

Foundations of Python Programming for Robotics Simulation

Python is a widely used text-based programming language known for its simplicity and readability, making it ideal for beginners. Beyond introductory use, it is a powerful tool for web development, software engineering, data analysis, and machine learning.

Before running Python programs, it is important to ensure that Python is installed on your computer. If it is not, you will need to download and install it.

Use the link or QR code below to watch a short video that guides you through checking your installation and setting up Python if necessary.

Link and Description	QR Code
Python Installation: Python Setup and IDE installation https://www.youtube.com/watch?v=CScxy0294SE&list=PLsyeobzWxl7poL9JTVyndKe62ieoN-MZ3&index=3	

There are several lightweight Integrated Development Environments (IDEs) available for writing and running Python programs. Common examples include PyCharm, IDLE (which comes bundled with Python), Thonny, Geany, and VS Code. If you have LEGO Education SPIKE installed, it also includes a built-in Python environment.

The SPIKE Prime Hub that you are familiar with is a small computer called a microcontroller, which has limited memory and processing power. Since the full Python programming language would use too much memory, the Hub runs MicroPython, a highly optimised version of the Python language that can run on microcontrollers.

In this session, we will use standard Python to build foundational skills. Once learners are comfortable with Python, transitioning to MicroPython will be straightforward.

Understanding Python Syntax

One of the first steps in learning Python is understanding its syntax which is basically the set of rules that define how code is written and structured. Python uses indentation (spaces at the beginning of a line) to group statements into blocks. All lines within the same block must have the same indentation level.

Each Python statement typically ends with a line break. Proper indentation is critical, as it defines the logic and flow of the program. Incorrect indentation will result in syntax errors.

Python programs are saved with the .py extension. When you run a .py file, your editor passes it to the Python interpreter which is a tool that reads the code and executes it line by line. For example, the command print("Hello") tells Python to display the word Hello on the screen. We will explore functions in more detail in a later lesson, but for now, note that functions are used to perform specific tasks. The print()function, for instance, displays whatever is placed within its brackets on the screen.

Most modern editors use syntax highlighting, which visually distinguishes different parts of the code. For instance:

1. Keywords like print may appear in blue.
2. Text inside quotation marks (strings) is often shown in orange.
3. Comments, which begin with #, appear in grey or green.
4. input() for collecting user input
5. () for calling functions

These elements make up the foundational structure of Python programs.

Now that the basics of syntax are understood, we can begin to work with the core constructs of text-based programming using Python.

Variables and Data Storage

Variables are named storage locations used to hold data. These are like variables in algebra. Once defined, variables can be used throughout a program to store and update values as needed. In LEGO SPIKE™ PRIME, block programming, variables are created when we want to store values which we would use later.

Figure 11.2: *Creating variables in block programming*

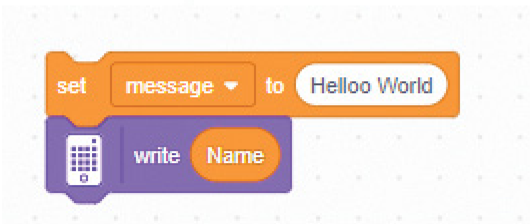
Figure 11.2 shows Variables being created and assigned values in block programming. Variables in Python function in a similar way. In Python, a variable is created by assigning a value to a name using the assignment operator =. ***For example:***

- robot_speed = 50
- distance = 20.5

Variable names should be descriptive and follow Python's naming conventions, typically lowercase letters with underscores separating words.

Variables are essential in robotics programming, where they can represent sensor readings, motor speeds, elapsed time, or user input. They make programs dynamic and adaptable, allowing values to change during execution.

Below is an example of how variables are created and used in block and python code.




```
message = "Hello World!"
print(message)
```

Figure 11.3: *Variable Usage in block and python programming*

In ***Figure 11.3***, the text "Hello world!" is stored in the variable message. When message is passed to the print() function, Python displays the value stored in the variable on a computer screen.

When using variables in Python, we need to adhere to a few rules and guidelines. Breaking some of these rules will cause errors; other guidelines just help us write code that is easier to read and understand. Be sure to keep the following variable rules in mind:

1. Variable names can contain only letters, numbers, and underscores. They can start with a letter or an underscore, but not with a number. For instance, you can call a variable *message_1* but not *1_message*.
2. Spaces are not allowed in variable names, but underscores can be used to separate words in variable names. For example, *greeting_message* works, but *greeting message* will cause errors.
3. Avoid using Python keywords and function names as variable names; that is, do not use words that Python has reserved for a particular programmatic purpose, such as the words print, is, class, return, none, pass etc.

4. Variable names should be short but descriptive. For example, *name* is better than *n*, *student_name* is better than *s_n*, and *name_length* is better than *length_of_persons_name*.
5. Be careful when using the lowercase letter l or uppercase I (as in **l**etter or Import) and the uppercase letter O because they could be confused with the numbers 1 and 0.

Basic Data Types

Data types classify the kind of values a variable can hold, influencing how the data is stored and manipulated. Fundamental Python data types relevant to robotics simulation include:

1. **Integers (int):** Whole numbers (e.g., 0, 10, -5). Used for counts, speeds, or discrete values.
2. **Floating-Point Numbers (float):** Numbers with a decimal point (e.g., 3.14, -0.5, 100.0). Used for distances, time durations, or sensor readings that require precision.
3. **Booleans (bool):** Represent truth values, either True or False. Used for logical conditions, sensor states (e.g., button pressed or not), or binary outcomes.
4. **Strings (str):** Sequences of characters enclosed in single or double quotes (e.g., "Hello World", "Motor A"). Used for displaying text, labelling outputs, or storing descriptive information.

Operations In Python

You can write mathematical expressions in python using common operators like add (+), subtract (-), divide (/), multiply (*), modulo (%) (remainder), exponent (**). ***Fig 4*** shows some operator blocks used in block programming as well as how they are used and the corresponding python code.

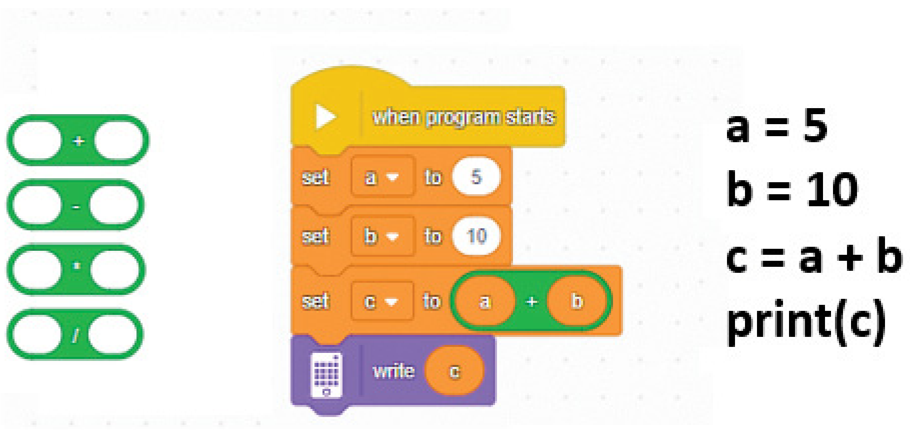


Figure 11.4: *Operators in Block programming and Python*

Figure 11.5 shows how various data types are used with the various operations. The "//" operator as shown is used to integer divide. It will remove all decimals.

```
>> print(5+10)
15
>> print(10/3)
3.3333333333333335
>> print(10//3)
3
>> print("ab"+"cd")
abcd
>> print(7+"ab")
TypeError: unsupported operand
type(s) for +: 'int' and 'str'
>> print(7,"ab")
7 ab
```

Figure 11.5: *Sample Python operations*

You can add numbers, floats, strings, and many more but you cannot interchange different types in operations. This is demonstrated from the Error message when we tried to print the results of adding two different types of data. The 10/3 output ends in a 5 because of something called "floating point approximation". Basically, computers have to estimate when decimals are involved, so there is some inaccuracy.

Operations involving variables are not all the same when compared to the branch of mathematics known as algebra. Expressions are right-hand evaluated meaning the expression on the right of the = is evaluated first, then re-casted to the variable on the left side

```
>> x = 10
>> print(x)
10
>> x = x+10
20
>> # Shorthand:
>> x+=10
30
```

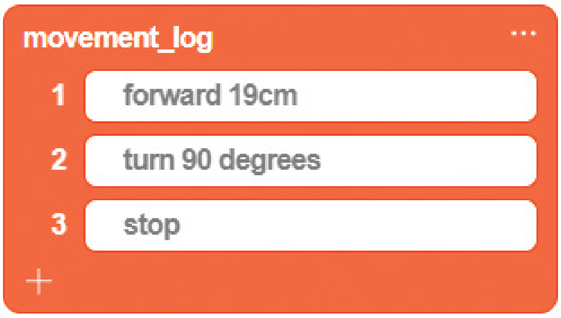
Figure 11.6: *Operations in Variables*

In ***Figure 5.6***, the x+10 is evaluated to 20 first, then x is set to 20, deleting the previous value.

Lists

Lists are ordered, mutable collections of items. They are highly versatile and can store sequences of values of different data types. In robotics, lists are invaluable for representing and managing collections of related data, such as:

1. **Movement Logs:** A list could store a sequence of robotic movements, each represented by a string or a tuple of parameters. Example:




```
movement_log = ["forward 10cm",
"turn 90deg", "stop"]
```

Figure 11.7: *Creating lists in Block and Python code*

2. **Sensor Input History:** A list can keep a record of sensor readings over time, enabling analysis of trends or past states. Example:
   - light_readings = [50, 52, 48, 55, 60]

3. **Robot State Parameters:** A list might hold various aspects of a robot's current state, such as its coordinates, battery level, and current task. Example:
   - robot_state = [10, 20, "moving forward", 75.5]

Lists support various operations, including adding items (append(), insert()), removing items (remove(), pop()), accessing items by index, and iterating through items.

For more on lists and how they are used in python programming follow the link below

Link	QR CODE
How to Use Lists in Python https://www.youtube.com/watch?v=9OeznAkyQz4	
Python Tutorial for Beginners 4: Lists, Tuples, and Sets https://www.youtube.com/watch?v=W8KRzm-HUcc	

Learning Tasks

This task requires the application of fundamental Python programming concepts to create a program that manages and displays simulated operational data for a robotic system. The aim is to demonstrate proficiency in basic Python syntax, variable management, data type utilisation, mathematical operations, and the implementation of lists.

Task Requirements

1. **Robot System Identification**
   - **a.** Define a string variable named robot_id and assign it a unique identifier for your simulated robot (e.g., "AccraBot-Alpha", "AgriMech-Beta").
   - **b.** Define a string variable named system_purpose and assign it a brief description of what your robot is designed to do (e.g., "Agricultural crop monitoring", "Urban sanitation support").
   - **c.** Output these identification details to the console using a clear, descriptive format.
2. **Operational Data Capture**
   - **a.** Define an integer variable named mission_duration_minutes to represent the elapsed time of a hypothetical mission (e.g., 120).
   - **b.** Define a floating-point variable named current_power_level_percent to represent the robot's current battery or fuel level (e.g., 78.5).
   - **c.** Define a Boolean variable named is_operational to indicate if the robot is currently functioning correctly (True or False).
   - **d.** Output each of these operational data points to the console with appropriate labels.
3. **Performance Calculation**
   - **a.** Using the mission_duration_minutes and an assumed average power consumption rate (e.g., 0.5% per minute), calculate the total_power_consumed_percent.
   - **b.** Calculate the estimated_remaining_operating_minutes based on the current_power_level_percent and the assumed power consumption rate.
   - **c.** Output both calculated values to the console, clearly stating what each number represents.
4. **Activity Log Creation**
   - **a.** Create a Python list named recent_activities to store a chronological record of the robot's last five actions or status updates. Populate this list with descriptive string entries (e.g., "Initiated drone deployment", "Detected irregular soil moisture", "Returned to charging station", "Completed urban route", "Performed data upload").
   - **b.** Output the entire recent_activities list to the console.
   - **c.** Output only the first and last activities from the recent_activities list to the console, each on a separate line.
5. **Consolidated Report Generation**
   - **a.** Combine all the outputs from the previous steps into a single, well-structured robot status report. Ensure the report is clearly organised, easy to read, and all information is accurately presented and labelled. The report should begin with a title, such as "ROBOT SYSTEM DAILY STATUS REPORT."

PEDAGOGICAL EXEMPLARS

Recognise that learners possess shared characteristics while also having individual differences in interests, readiness, and learning styles. Plan to address these variations. Provide multiple avenues for learners to engage with the lesson content, adjusting the level of detail or support as needed to help all learners achieve the lesson outcomes. Employ strategies such as Think-Pair-Share to encourage participation. Before starting the lesson, the following are some advanced preparations that need to be made:

a. Ensure all classroom computers have Python, and a suitable Integrated Development Environment (IDE) installed and functioning correctly (e.g., Thonny, VS Code with Python extension, or the built-in LEGO SPIKE™ Prime Python editor).

b. Prepare clear visual aids: large-font examples of Python syntax (e.g., print() function, variable assignment, list creation), a chart displaying common data types, and a sample output format for the robot status report.

c. For differentiation, prepare simplified code templates with comments for learners who may require more structure, and challenge questions for those who grasp concepts quickly.

1. Step 1: Recap and Introduction to the Python Environment

Begin by briefly reviewing the shift from block-based programming to text-based Python, reiterating its advantages for complex robotics. Introduce the chosen Python IDE, demonstrating how to open a new Python file and execute a simple print() statement. Emphasise the critical role of indentation in Python syntax.

a. Differentiation: For learners who are new to text editors, provide a quick guided tour of the IDE's interface, highlighting key areas (editor window, output console, run button). For advanced learners, briefly discuss keyboard shortcuts for common actions.

b. Inclusivity: Verbally describe each step of navigating the IDE and typing code. Ensure the projected display is high-contrast and font size is adequate. Offer a printed checklist for environment setup for learners who benefit from visual aids.

2. Step 2: Understanding Variables and Data Types (Addressing Task Requirements 1 & 2)

Guide learners in defining variables and understanding data types. Explain how variables act as containers for different kinds of robot data (e.g., robot name, power level). Demonstrate how they are created in block programs using familiar LEGO SPIKETM PRIME software and link them to how variables are created in Python. Demonstrate how to assign values and use the print() function to display variable content with descriptive labels.

a. Differentiation

i. For Learners Needing Support: Provide pre-written variable names and initial values for them to input, focusing on the print() statements. Offer a "fill-in-the-blanks" activity for their initial code.

ii. **For Advanced Learners:** Challenge them to use the input() function to allow the user to provide the robot's ID or purpose, introducing user interaction.

b. **Inclusivity:** Use tangible analogies for variables (e.g., labelled boxes, folders). Use colour-coding in code examples (syntax highlighting) to aid readability. Encourage peer pairing to facilitate collaborative learning and problem-solving.

3. **Step 3: Mastering Operations and Lists (Addressing Task Requirements 3 & 4)**

Facilitate the understanding of arithmetic operations for calculations, such as determining power consumption or estimated operating time. Introduce Python lists as a method for storing ordered collections of data, such as a log of robot activities. Demonstrate list creation, adding elements, and accessing specific elements using indices.

a. **Differentiation**

i. **For Learners Needing Support:** Provide the exact mathematical formulas for calculations and pre-defined lists for them to manipulate. Focus on the correct application of operators and list methods (.append()).

ii. **For Advanced Learners:** Encourage them to consider additional calculations (e.g., average speed) or implement more complex list manipulations (e.g., removing specific items, sorting the list).

b. **Inclusivity:** Verbally explain the sequence of operations in calculations (order of operations). Use concrete examples for lists, such as "a list of chores" or "a list of favourite foods" before moving to robot activities. Provide step-by-step written instructions for list methods.

4. **Step 4: Consolidating and Reporting (Addressing Task Requirement 5)**

Guide learners in combining all elements from the previous sub-tasks into a single, cohesive robot status reporter program. Emphasise the importance of clear output formatting, logical organisation of information, and descriptive labels within their print() statements to create a professional-looking report.

a. **Differentiation**

i. **For Learners Needing Support:** Provide a basic template for the overall report structure, with placeholders for their variables and lists. Offer guidance on using commas within print() statements to combine text and variables effectively.

ii. **For Advanced Learners:** Challenge them to implement more advanced formatting techniques (e.g., f-strings for cleaner output), or to consider how the report might be saved to a simple text file rather than just printed to the console.

b. **Inclusivity:** Encourage learners to verbalise their thought process for structuring the report before coding. Allow for flexibility in the exact presentation format as long as the content requirements are met. Provide examples of clear, well-formatted reports.

5. **Step 5: Review and Troubleshooting**

   Dedicate time for learners to test their programs, identify errors, and debug. Guide them through common Python errors (e.g., SyntaxError, NameError, TypeError, IndentationError), teaching them how to read error messages and trace problems in their code. Conclude by having learners share their functional status reporters.

   a. **Differentiation**

      i. **For Learners Needing Support:** Provide common error messages and their typical causes as a troubleshooting guide. Offer direct, one-on-one support for debugging. Encourage them to use print() statements strategically to inspect variable values at different points.

      ii. **For Advanced Learners:** Challenge them to identify and explain complex errors in provided flawed code snippets or to assist peers in debugging.

   b. **Inclusivity:** Emphasise that encountering errors is a natural part of programming and a learning opportunity. Foster a supportive environment where learners feel comfortable seeking help and learning from mistakes. Provide visual examples of common error messages and their solutions.

KEY ASSESSMENT LEVEL 4: INDIVIDUAL PRACTICAL TEST AND PRESENTATIONS

DoK Level 3: Short Answers and Extended Response Essays - (Suggested Modes: Individual Classwork/Homework, Case Study) - *20 Marks*

1. **Data Management Strategy for a Humanitarian Robot**

   A newly designed robotic system, intended for delivering medical supplies to remote villages in Ghana during emergencies, needs to manage critical operational data effectively. The robot must log its route points, track its payload status, and record any environmental anomalies (e.g., unexpected rainfall, rough terrain) encountered.

Instructions: Write a detailed response that addresses the following:

   a. **Justify the transition from block-based programming to Python for this specific humanitarian robot application.** Provide at least two distinct reasons directly relevant to the complexity or requirements of such a robot. (4 Marks)

   b. **Propose a comprehensive Python variable and data type strategy** for storing the following pieces of information for the robot:

      i. Robot identification number (e.g., "DRONE-GH-001")

      ii. Current payload weight (in kilograms, which might not be a whole number)

      iii. The total number of successful deliveries made

      iv. A status indicating if its main delivery hatch is open or closed

      v. A list of the last three critical events encountered during a mission (e.g., "Heavy Rainfall Detected", "Path Obstruction", "Low Battery Warning").

For each piece of information, state the most appropriate Python data type and provide a justification for your choice. (8 Marks)

2. **Explain how Python's list data structure would be specifically advantageous** for managing the last_three_critical_events compared to storing each event in a separate, individual variable. (6 Marks)

DoK Level 4: Individual Practical Task (Mode: Individual Presentation)

Advanced Robot Data Analyser (*18 Marks plus 2 bonus marks*)

Task: Expand upon the "Robot System Status Reporter" program created in the learning tasks or develop a new Python program that simulates a robot collecting continuous sensor data and performing basic analysis.

Instructions

1. **Data Generation & Storage (6 marks plus 2 for logical alignment and syntax):**
   - **a.** Create a Python list named temperature_readings and populate it with at least ten simulated temperature values (floating-point numbers, representing degrees Celsius) over time.
   - **b.** Create another list named humidity_readings with ten corresponding simulated humidity values (integers, representing percentages).
   - **c.** Simulate a robot's location data by creating a list of tuples, robot_path, where each tuple (x, y) represents a coordinate pair (e.g., [(0,0), (1,2), (3,1), ...]). This list should contain at least five distinct points.
2. **Data Analysis & Calculation (5 marks plus 2 bonus marks for part iv):**
   - **a.** Calculate the average temperature from the temperature_readings list.
   - **b.** Calculate the maximum humidity from the humidity_readings list.
   - **c.** Using Python's arithmetic operations, determine the total 'distance' travelled by the robot, assuming each step between consecutive points in robot_path is 1 unit. (Hint: count the number of steps in the path).
   - **d.** (Optional for higher challenge): Calculate the Euclidean distance between each consecutive point in robot_path and sum them up for the total distance.
3. **Structured Output (5 marks)**
   - **a.** Output a comprehensive "Robot Environmental and Path Analysis Report" to the console. This report should clearly present:
     - The original lists of temperature_readings, humidity_readings, and robot_path.
     - The calculated average temperature.
     - The calculated maximum humidity.
     - The total 'distance' travelled by the robot.
   - **b.** Ensure the report is well-formatted, uses clear labels, and is easy to interpret. (5 Marks)

Conclusion

This section presents a foundational shift from block-based programming to Python, equipping learners with the precision and scalability required for advanced robotics. Through structured exposure to syntax, variables, and data types, students begin to craft functional programs capable of managing robotic behaviour. The use of real-world tools like Pybricks and Thonny prepares learners for professional development environments.

FOCAL AREA 2: UNDERSTANDING CONDITIONAL STATEMENTS

This session introduces learners to conditional statements and looping constructs, fundamental elements for enabling robotic systems to make decisions and perform repetitive tasks. The lesson elaborates on simple if statements, if-else structures, and if-elif-else chains, demonstrating how this control the program flow based on various conditions. Furthermore, it covers while and for loops, crucial for automating actions and managing tasks with known or unknown repetitions. The section also addresses loop control statements like break and continue, which provide precise control over loop execution. This understanding is vital for developing sophisticated robotic behaviours that respond dynamically to environmental feedback and sensor inputs.

Building on Foundational Concepts

Building on foundational concepts from Year 1, such as implementing logic and loop diagrams (Week 5) and creating programs with single and nested decision conditions (Weeks 21 - 23), Focal Area 2 advances learners to text-based Python programming. This section deepens their understanding of control flow by translating previously visual block-based logic into precise Python syntax for conditional statements and loops. Concepts like Boolean Algebra (Year 2, Week 9) become critical for defining robust conditions, enabling learners to develop more sophisticated and flexible robotic behaviours beyond foundational visual programming.

Recap Weeks

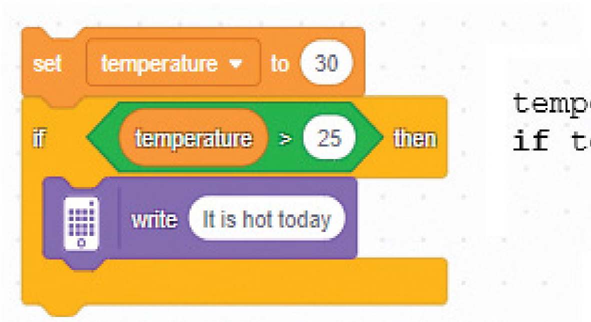
Year 1	**Week 5** • Evaluate the use of logic and loop diagrams and demonstrate their use in control systems' design. **Weeks 21 - 23** • Create computer programs from pre-designed flowcharts that have single-decision conditions • Create computer programs from pre-designed flowcharts that have nested decision conditions. **Weeks 24** • Create computer programs from pre-designed flowcharts that have a controlled feedback loop with loop-interrupts

Year 2	**Week 5** • Analyse and derive logical models for the implementation of finite state machines. **Week 9** • Learners should explain and apply Boolean Algebra in the definition and design of digital systems. **Week 17** • Learners should explain and apply Boolean Algebra in the definition and design of digital systems. **Week 23 and 24** • Learners should observe and identify flaws in an operational robot and trace them to either algorithm flaws, design flaws or coding errors. • Learners should use series of iterations in fixing the identified flaws or improve performance of solutions.

KNOWLEDGE CONTENT

Conditional statements allow a program to make decisions based on conditions. The if statement is used when an action should be performed only if a specific condition is true. In year one, you learnt about the different **If** blocks that exist in Block programming and how they are used. It works in the same way, but they are expressed differently.

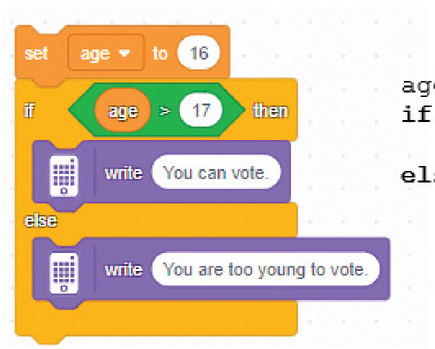
1. **Simple if Statements**: These are used when a single condition is checked, and a corresponding block of code is executed only if the condition holds.




```
temperature = 30
if temperature > 25:
    print("It is hot today.")
```

Figure 11.8: *Simple if statements*

2. **If-else Statements**: These provide an alternative action when the condition is false. ***Example***:




```
age = 16
if age > 17:
    print("You can vote.")
else:
    print("You are too young to vote.")
```

Figure 11.9: *if-else Statements*

3. **if-elif-else Chains**: These allow multiple conditions to be checked sequentially. This only exist in python code, but it functions like the **if-else Statements** but has more **else** conditions. ***Example***:

```
traffic_light = "yellow"
if traffic_light == "green":
    print("Go")
elif traffic_light == "yellow":
    print("Slow down")
else:
    print("Stop")
```

Such control structures are essential in robotics where robots must react differently based on sensor inputs or environmental feedback.

Nested Decision Conditions

Nested decision conditions refer to placing one if, elif, or else block inside another. This is useful when decisions depend on multiple layers of conditions — like in robotics, where a robot might first check if a sensor is active and then make decisions based on the sensor value.

```
if condition1:
    # Outer if block
    if condition2:
        # Inner if block
        # Do something if both condition1 and condition2 are True
    else:
        # Do something if condition1 is True but condition2 is False
else:
    # Do something if condition1 is False
```

Example **1: Battery Level and Movement Decision**

```
battery_level = 60
obstacle_detected = True

if battery_level > 50:
    print("Battery level sufficient.")
    if obstacle_detected:
        print("Stop and reroute.")
    else:
        print("Continue moving forward.")
else:
    print("Battery too low! Return to charging station.")
```

First, the robot checks if the battery is sufficient.

1. If yes, it checks for an obstacle.
2. If an obstacle is detected, it stops.
3. If not, it moves forward.
4. If the battery is not sufficient, it returns to charge.

Best Practices When Using Nested Conditions

1. Keep indentation clear (use 4 spaces or a tab).
2. Avoid nesting too deeply — refactor into functions when possible.
3. Combine related conditions using and/or if it keeps things simpler.

For more on if-else statements and how they are used in Python, follow the link below:

Link	QR Code
If Else Statements in Python https://www.youtube.com/watch?v=-BOBedcjySI	
Conditionals and Booleans - If, Else, and Elif Statements https://www.youtube.com/watch?v=DZwmZ8Usvnk	

Looping Constructs

Loops are used to repeat blocks of code. The concept of looping and the various types of blocks used to loop in block programming has been considered in Year one. This section will focus on two main types of looping constructs: **while** and **for** Loops.

1. **while** Loops: Run as long as a condition remains true. ***Example***:

```
count = 1
while count <= 5:
    print(count)
    count += 1
```

The code demonstrates how a while loop repeatedly runs a block of code as long as a given condition is true. It starts by initialising a counter, checks if the counter is less than or equal to 5, prints it, and increases the counter by 1. This continues until the condition becomes false, at which point the loop stops. The example highlights the need for a condition, an update step, and a clear stopping point to avoid infinite loops. This is useful when the number of repetitions is not predetermined, such as in sensor monitoring tasks.

2. **for Loops**: Used for iterating over sequences (lists, ranges, strings). Example:

```
for number in range (1, 6):
    print(number)
```

The code uses a for loop to iterate over a sequence of numbers generated by range(1, 6), which includes 1 through 5. In each loop cycle, the variable number takes the next value in the sequence. The print(number) line outputs the current number to the screen. This example shows that for loops are ideal for repeating actions a known number of times without manually updating a counter. This loop is particularly helpful for repeating tasks with a known number of repetitions, such as moving a robot forward five times.

Loop Control Statements

1. **break Statement**: Immediately stops the loop when a condition is met. ***Example***:

```
while True:
    command = input ("Enter command: ")
    if command == "stop":
        break
```

2. **continue Statement**: Skips the current iteration and proceeds to the next one. ***Example***:

```
for num in range (10):
    if num % 2 == 0:
        continue
    print(num)
```

3. **Avoiding Infinite Loops**: Infinite loops occur when the terminating condition is never met. Care must be taken to ensure that the loop has a proper exit strategy. ***Example***:

```
counter = 0

while counter < 5:
    print ("Counting:", counter)
    # counter is never incremented, so the condition remains true forever
```

Nested Loops in Python

A nested loop is when one loop is placed inside another loop. In robotics, this is useful for tasks like grid navigation, sensor calibration, scanning multiple zones, or controlling multi-component systems (e.g., motors and servos working together).

Basic Structure of Nested Loops

```
for outer in range (3):
    print ("Outer loop:", outer)
    for inner in range (2):
        print (" Inner loop:", inner)
```

How Robots Use Nested Loops

Scenario	What Nested Loops Do
Scanning a 2D grid	Outer loop = rows, inner loop = columns
Testing multiple sensor-motor combinations	Outer = sensors, inner = motor speeds
Line-following with pattern detection	Outer = lines, inner = pattern on each line

Example 1: **Robot Scanning a 3 x 3 Grid**

```
for row in range (3):
    for col in range (3):
        print (f"Scanning cell at Row {row}, Column {col}")
```

Output

- Scanning cell at Row 0, Column 0
- Scanning cell at Row 0, Column 1
- Scanning cell at Row 2, Column 2

Example 2: **Robot Testing Multiple Sensor Thresholds**

```
sensor_thresholds = [10, 20, 30]
motor_speeds = [50, 100]

for threshold in sensor_thresholds:
    for speed in motor_speeds:
        print (f"Testing with sensor threshold {threshold} and motor speed
{speed}")
```

Example 3: **Nested while Loops – Wait for Obstacle to Clear**

```
Obstacles = [[True, False], [False, False]] # 2x2 grid


row = 0
while row < len(obstacles):
   col = 0
   while col < len(obstacles[row]):
      if obstacles[row][col]:
         print (f"Obstacle at ({row}, {col})! Waiting...")
      else:
         print (f"Path clear at ({row}, {col}). Proceed.")
      col += 1
   row += 1
```

Tips for Working with Nested Loops:

1. ***Keep loops simple:*** Avoid nesting more than 2–3 levels deep.
2. Use meaningful variable names (e.g., row, col, sensor, speed).
3. Consider using functions to make code cleaner and reusable.

Note: Be careful with break or continue — they only affect the current loop unless clearly managed.

Challenge Task for Students (Grid Navigation Simulation)

Task: Write a program that simulates a robot scanning a 5×5 floor grid. At each cell, the robot should:

1. Print the current location.
2. If the sum of the row and column is divisible by 4, mark the cell as "High traffic".

These concepts are particularly relevant when programming robots to respond to events or repeatedly check for certain sensor values until a condition is met. Block programming also has different blocks that work like these looping constructs like the "repeat until" block, "wait until" block and "forever" blocks.

For more on loops and how they are used in Python, follow the following link:

Link	QR Code
Loops and Iterations – For/While Loops https://www.youtube.com/watch?v=6iF8Xb7Z3wQ	

Practical Relevance to Robotics

In robotics, control flow is the backbone of autonomous behaviour. We will need to use if statements to guide robot decisions based on sensor input (e.g., detecting obstacles), and loops to repeat behaviours like checking line-following sensors or executing repetitive movements. Understanding how and when to stop or skip code execution will help learners debug and write efficient robot control scripts.

Learning Tasks

Develop a Python script that guides a simulated robot through a series of actions based on environmental conditions and repetitive routines.

Sub-tasks:

1. Temperature-Based Movement:
   a. Create a program that prompts for a simulated "room temperature" as an integer input.
   b. Implement conditional logic using if, elif, and else statements to determine the robot's action:
      - If the temperature is above 25 degrees Celsius, the robot should print: "Temperature is high. Robot is moving to a cooler area."
      - If the temperature is between 18 and 25 degrees Celsius (inclusive), the robot should print: "Temperature is optimal. Robot is maintaining current position."
      - If the temperature is below 18 degrees Celsius, the robot should print: "Temperature is low. Robot is seeking a warmer area."
2. Repetitive Scanning Routine:
   a. Implement a for loop to simulate the robot performing a "scan" action five times.
   b. **Within the loop, print:** "Performing scan operation... [Current Scan Number]".
   c. Introduce a condition where, if the scan number is an even number, the robot prints: "Skipping detailed analysis for this scan." using a continue statement.
3. Emergency Stop Sequence:
   a. Design a while loop that continuously prompts the user to "Enter command:".
   b. If the user inputs "stop", the robot should print: "Emergency stop initiated. Robot shutting down." and the loop must terminate immediately using a break statement.
   c. For any other command, the robot should print: "Command received: [User Command]. Robot continues operation."

Ensure the program executes sequentially, demonstrating each behaviour as specified.

PEDAGOGICAL EXEMPLARS

This pedagogical approach focuses on guiding learners through the practical application of conditional statements and looping constructs in Python to simulate robotic behaviours. It acknowledges diverse learning needs and promotes a clear understanding of control flow in programming.

1. **Step 1: Recalling Foundational Concepts**

   Begin by briefly reviewing the theoretical aspects of conditional statements (if, else, elif) and looping constructs (for, while), as well as loop control statements (break, continue). Connect these to similar block programming concepts they learnt in Year one and how the logic worked. Connect these programming concepts directly to the necessity of robots making decisions and performing repetitive actions autonomously. Introduce the "Robotic Control Logic Challenge," emphasising that the focus is on applying these recently acquired programming skills to create functional simulated robotic responses.

   - **Differentiation and Inclusivity:** Display visual representations of control flow (e.g., simple flowcharts for if-else or for loops) from prior lessons. State the learning objectives for the challenge verbally and in written form, ensuring accessibility for all learners.

2. **Step 2: Structuring Conditional Behaviour**

   Guide learners in initiating the Python program. Direct them to implement the first sub-task, "Temperature-Based Movement," which requires using if, elif, and else statements. Provide a basic input mechanism for the simulated temperature and then facilitate the correct placement of conditional checks to trigger distinct robot responses. The emphasis here is on precision in logical conditions and understanding how different temperature ranges dictate the robot's simulated action.

   - **Differentiation and Inclusivity:** Offer a skeleton code structure with placeholders for the conditional logic for learners requiring additional support. For those who demonstrate readiness, encourage them to consider adding more granular temperature ranges or integrating multiple environmental factors into the decision-making process. Ensure clear instructions are presented both visually and audibly.

3. **Step 3: Orchestrating Repetitive Actions and Control**

   Facilitate the implementation of the "Repetitive Scanning Routine" and the "Emergency Stop Sequence." For the scanning routine, guide learners in using a for loop to manage a fixed number of repetitions and the continue statement to skip specific iterations based on a condition. Subsequently, direct them to construct the emergency stop feature using a while loop, focusing on how user input can trigger immediate loop termination via the break statement. This step highlights the power of iteration and explicit control over loop execution.

   - **Differentiation and Inclusivity:** Provide working examples of simple for and while loops for reference. Suggest common pitfalls, such as infinite loops, and prompt learners to think about how to avoid them. Challenge advanced learners to implement a counter within the while loop or to introduce a time delay in the for loop to simulate realistic scan durations. Encourage collaborative problem-solving among peers to address coding challenges.

4. **Step 4: Validating Program Functionality**

   Emphasise the critical importance of testing the complete program. Instruct learners to run their scripts multiple times with varied inputs for temperature and commands to ensure all conditional paths and loop behaviours function as intended. Guide them in systematically identifying and correcting any logical errors, syntax issues, or unintended outputs. This stage reinforces debugging as an integral part of the development process.

   - **Differentiation and Inclusivity:** Provide a simple checklist of test cases for each sub-task. Encourage peer review of code, fostering a collaborative learning environment where learners can assist one another in identifying discrepancies. Offer readily available resources for common Python error messages to support independent troubleshooting.

5. **Step 5: Articulating the Solution**

   Conclude by having learners demonstrate their simulated robot's behaviour. Encourage them to explain their code logic, specifically how conditional statements and loops enable the robot to respond dynamically to simulated environmental data and user commands. This final phase allows learners to articulate their understanding of control flow and its practical application in robotics.

   - **Differentiation and Inclusivity:** Allow for various modes of demonstration, such as a live execution of the code with verbal explanation, or a pre-recorded video of the simulation with voice-over commentary. For learners who may experience communication difficulties, offer the option of a written explanation accompanying their code demonstration. Ensure all learners have the opportunity to present their work and receive constructive feedback.

KEY ASSESSMENT LEVEL 4: SHORT ANSWER AND EXTENDED RESPONSE ESSAYS

(Suggested Mode: Case Study)

1. **Enhancing Robot Line-Following Resilience (25 Marks)**

   Analyse an existing robot's line-following code and propose modifications to enhance its resilience to common environmental challenges, specifically focusing on the application of advanced conditional statements and looping constructs.

   Scenario: A basic line-following robot uses a single sensor that returns a value: 0 for "on the line" and 1 for "off the line." It attempts to stay on the line.

```
def get_line_sensor_reading():
    # Simulates reading; returns 0 (on line) or 1 (off line)
    # For testing, assume it might return 1 randomly sometimes.
    return 0 # Or simulate more complex behavior for testing

def turn_left():
    print("Turning Left to find line")

def turn_right():
    print("Turning Right to find line")

print("Starting Line Following")
while True:
    sensor_value = get_line_sensor_reading()
    if sensor_value == 0:
        print("Moving Straight: On Line")
    else: # sensor_value == 1
        print("Adjusting: Off Line")
        # Simplified adjustment - needs improvement
        turn_left() # Placeholder for actual turn logic
```

New Requirement

The robot needs to be more robust. If it goes off the line (sensor_value == 1), it should not immediately give up. Instead, it should initiate a "search pattern" for the line. This search pattern involves:

1. Attempting to turn_left() and check the sensor.
2. If still off the line after turn_left(), it should turn_right() and check the sensor.
3. This entire turn_left, turn_right sequence (one full search attempt) should be repeated a maximum of 3 times.
4. If, after 3 full search attempts, the line is still not found, the robot should report_line_lost() and halt_operation().

Tasks

1. **Code Modification Proposal (*10 Marks*)**
   - **a.** Write a detailed proposal for how you would modify the Current Simplified Code Snippet to implement the New Requirement. Your proposal should include specific Python code segments for the search pattern and the logic for reporting a lost line and halting operation.
   - **b.** Clearly indicate where if-elif-else chains, while loops, for loops, break, and continue statements would be used in your proposed solution.

2. **Justification of Control Structures (*10 Marks*)**
   - Justify your choice of specific control structures (e.g., why a while loop over a for loop for the search attempts, or vice versa; why break or continue at certain points) for each part of the New Requirement's logic. Explain how these choices contribute to the robot's efficient and robust behaviour.

3. **Testing Strategy (*5 Marks*)**

   a. Outline a strategy for how you would test your modified code to ensure that:

      i. The robot successfully re-acquires the line after being briefly off.

      ii. The robot correctly executes the full 3 search attempts if the line is not found.

      iii. The robot accurately report_line_lost() and halt_operation() if the line is not re-acquired after 3 attempts.

Conclusion

This focal area introduces conditional logic and iterative control in Python. Learners are guided to construct branching decisions and looping mechanisms, essential for automating robot responses. Practical applications like environmental sensing and real-time feedback systems are emphasised to ensure learners can build responsive robotic behaviours grounded in logic and structure.

FOCAL AREA 3: FUNCTIONS AND THEIR ROLE IN ROBOTICS PROGRAMMING

This lesson introduces the fundamental concept of functions, essential tools for structuring code effectively. Learners will understand how to define custom functions, pass information through parameters, and retrieve results using return values. This approach, known as modular programming, allows complex robotic tasks to be broken down into smaller, reusable components, significantly enhancing program readability and simplifying the debugging process. By mastering functions, programmers can build more organised, efficient, and scalable control systems for autonomous robots, promoting a systematic approach to complex challenges.

Building on Foundational Concepts

This section significantly builds upon foundational programming concepts from previous years. Learners previously designed control system logic using diagrams and structured problems with algorithms, pseudocodes, and flowcharts (Year 1 & 2, Week 17). Functions provide the formal Python mechanism to encapsulate these defined 'processes,' enabling true modular programming. Furthermore, the understanding of conditional logic (Year 1, Weeks 21-23) and Boolean Algebra (Year 2, Week 9) becomes vital for defining precise function behaviours. This progression allows for creating highly organised, reusable code, essential for developing complex, menu-driven robotic control systems and elevating the ability to tackle sophisticated challenges.

Recap Weeks

Year 1	**Week 5** • Evaluate the use of logic and loop diagrams and demonstrate their use in control systems' design. **Week 17** • Determine the Inputs, Processes and Outputs required to solve a particular problem. • Define solutions to basic automated and robotic problems using algorithms, pseudocodes, and flowcharts diagrams. **Weeks 21 - 23** • Create computer programs from pre-designed flowcharts that have single-decision conditions • Create computer programs from pre-designed flowcharts that have nested decision conditions. **Weeks 24** • Create computer programs from pre-designed flowcharts that have a controlled feedback loop with loop-interrupts.
Year 2	**Week 5** • Analyse and derive logical models for the implementation of finite state machines. **Week 9** • Learners should explain and apply Boolean Algebra in the definition and design of digital systems. **Week 17** • Learners should explain and apply Boolean Algebra in the definition and design of digital systems.

KNOWLEDGE CONTENT

In Python, a function is a named block of code designed to perform a specific task. Instead of repeating code every time a robot must perform a repeated action like moving forward or turning. A function can be written once and reused throughout the program. This reduces errors, improves clarity, and makes troubleshooting easier.

An example is a simple function that prints a greeting message. This function is defined using the def keyword and is later called by its name.

```
def greet():
    print("Hello, Robot")


greet()
```

In a robotics context, similar functions can be written to start motors, stop the robot, or check sensors.

Functions with Parameters and Return Values

Functions can also accept input values (called *parameters*) that influence how they behave. This allows for greater flexibility in controlling different aspects of the robot's actions.

***Example*: A function with a parameter to control movement steps**

This function accepts the number of steps as input and uses it to print the movement instruction.

```
def move_forward(steps):
    print("Moving forward", steps, "steps")

move_forward(5)
```

In a real robot, the parameter steps could be used to determine how far the robot moves.

A function that returns a value: Some functions may need to send back a result, such as a sensor reading or a calculated value.

```
def get_speed():
    return 10

speed = get_speed()
```

Return values allow your robot's code to respond dynamically to changes in the environment or internal settings.

Using Dictionaries to Map Commands

A dictionary in Python is a set of key-value pairs. It can be used to create flexible control systems where a user command (like "left" or "right") maps directly to a robot function.

***Example*: Mapping text commands to robot actions using a dictionary**
In this example, two functions simulate robot actions. A dictionary connects command strings to these functions.

```
def turn_left():
    print("Turning left")

def turn_right():
    print("Turning right")

commands = {
```

```
    "left": turn_left,
    "right": turn_right
}

command = input("Enter command: ")
if command in commands:
    commands[command]()
else:
    print("Invalid command")
```

Receiving and Handling User Input

Python's input() function lets users type commands or values while the program is running. This is useful in robotics when simulating control through manual input or configuring settings during execution.

***Example*: Capturing and printing user input**

This program asks for the robot's name and greets it using the given input.

```
name = input("Enter your robot's name: ")
print("Hello", name)
```

This basic input method can be expanded to select modes, adjust speed, or provide simulated sensor values.

Writing Menu-Based Programs

Menu-based programs give users multiple options and allow them to choose how the program behaves. In robotics, this is used to let an operator select between tasks such as start, stop, or change mode.

***Example*: A simple command menu using a loop and functions**

This program presents a menu repeatedly until the user chooses to exit. Each choice triggers a different robot action.

```
def start ():
        print ("Robot started")

def stop ():
        print("Robot stopped")
```

```
def show_menu():
    print("1. Start\n2. Stop\n3. Exit")

while True:
    show_menu()

    choice = input("Select an option: ")
    if choice == "1":
        start()
    elif choice == "2":
        stop()
    elif choice == "3":
        break
    else:
        print("Invalid choice")
```

This structure allows for clear, interactive control systems that are easy to extend and modify.

For more on functions and how they are used in Python, follow the following link:

Link	QR Code
Functions https://www.youtube.com/watch?v=9Os0o3wzS_I	

Benefits of Code Structuring in Robotics

Using functions, dictionaries, and input-based menus makes a robot's program easier to manage. Structured code allows:

1. Clear separation of tasks
2. Easy updates and improvements
3. Simplified troubleshooting and testing
4. Reuse of code for similar robotic tasks

In practical robotics, these methods improve both the reliability and scalability of the system. As learners advance, these principles prepare them for more complex programming tasks in real-world automation projects.

ROBOTICS CHALLENGE: PRIME SCANNER BOT

Scenario

You are part of a team designing an intelligent robot tasked with sorting numbered packages on a conveyor belt. Some of these packages are labelled with positive integers, and your robot must scan the label and determine whether the number is prime. If it is prime, the robot routes it to a "Special Handling" area. Otherwise, it sends the package to "Standard Processing."

Task

Design and code a Python function that simulates the logic your robot would use to decide if a scanned number is prime. You will apply the concepts you've learned:

1. Variables (to store values like the number and counter)
2. Control structures (e.g., if, for, while)
3. Functions (to encapsulate the logic neatly)

Specifications

1. The function should be called is_prime(n) and take a positive integer n as input.
2. The function should return True if the number is prime, and False otherwise.
3. The main program should simulate the robot receiving numbers and making routing decisions based on the result.

***Example* Output**

Enter number on package: 7

Prime detected! Route package to Special Handling.

Enter number on package: 12

Not prime. Send package to Standard Processing.

Hints

- *A number is prime if it has no divisors other than 1 and itself.*
- *Check divisibility from 2 to n−1 (or more efficiently, up to int(n**0.5) +1).*
- *Use the modulo operator % to check for divisibility.*

Extension Task (Optional)

Simulate scanning multiple packages using a while or for loop. The robot should keep asking for input until the user enters 0 (zero) to stop.

Learning Tasks

Your task is to apply your understanding of functions, user input, and program structuring to create Python programs that simulate robotic behaviours.

1. Developing Core Robotic Functions
    a. Define two distinct Python functions that simulate basic robot actions. For example, move_forward(distance) to print a message about moving a given distance, and turn_robot(angle) to indicate a turn by a specified angle. Ensure these functions accept parameters.
    b. Create a function named get_sensor_reading() that simulates reading a sensor. This function should return a numerical value representing a sensor input, such as a simulated temperature or light level.
2. Interactive Robot Control with Dictionaries
    a. Implement a dictionary that maps specific text commands (e.g., "start", "stop", "status") to the functions you defined above.
    b. Write a program segment that prompts the user to enter a command. Use the input() function for this purpose.
    c. Utilise the created dictionary and user input to call the corresponding robot action function. Include an if-else statement to handle cases where the user enters an invalid command.
3. Building a Menu-Driven Robot Interface
    a. Design a menu-based control program for a simulated robot. This program should display a list of options (e.g., "1. Patrol Mode", "2. Recharge", "3. Exit").
    b. Implement this menu using a while loop, ensuring it continues to display until the user selects the "Exit" option.
    c. Each menu option should correspond to a unique function that performs the simulated robotic action.
    d. Incorporate conditional statements (if-elif-else) to process the user's choice and execute the correct function.
4. Prime Number Checker Challenge
    a. Write a Python program that determines if a given positive integer is a prime number.
    b. Define a function, is_prime(n), which takes an integer n as input. This function should return True if n is prime and False otherwise.
    c. Within your is_prime function, apply conditional statements and a loop to check for divisibility by numbers between 2 and n-1. Remember that prime numbers are only divisible by 1 and themselves. The modulo operator (%) will be useful here.

d. Prompt the user to enter an integer and use your is_prime function to print whether the number is prime or not.

PEDAGOGICAL EXEMPLARS

This pedagogical exemplar outlines a structured approach for delivering the lesson on functions in robotics programming, ensuring that all learners can engage meaningfully with the content.

1. **Step 1: Introduction to Functions as Modular Building Blocks**

   Begin by briefly reviewing the importance of structured code and control flow, concepts addressed in previous lessons. Introduce functions as self-contained blocks of code designed to perform specific tasks, emphasising their role in promoting code reusability and simplifying complex programs. Illustrate how functions can make robot control more organised and manageable.

   a. **Talk for Learning Strategy**: Use a Think-Pair-Share approach where learners first consider, then discuss with a partner, how using functions could help make a robot program easier to manage.

   b. **Experiential Strategy**: Guide learners to examine a simple block-based robot task, then collectively translate it into a basic Python function.

   c. To ensure differentiation and inclusivity, use simple, relatable analogies such as a set of instructions for assembling a toy or a specialised tool in a toolbox. Provide visual representations of function definitions and calls to support diverse learning styles.

2. **Step 2: Guided Construction of Fundamental Robotic Functions**

   Guide learners in the initial construction of basic Python functions as per Learning Task 1. Provide clear examples demonstrating how to define functions that accept parameters and return values, simulating fundamental robot actions like movement or sensor readings. Learners should practise defining functions such as move_forward(distance) or get_sensor_reading().

   a. **Talk for Learning Strategy**: In pairs, learners explain the purpose and structure of their functions to one another before running the code. Partners provide feedback or identify unclear naming or logic.

   b. **Experiential Strategy**: Each learner writes and tests at least two distinct robot control functions, observing how parameter values influence output.

   c. For learners requiring additional support, provide partial code structures to complete. Challenge those with greater aptitude to design functions with multiple interdependent parameters or more complex return types.

3. **Step 3: Implementing Interactive Robot Control**

   Facilitate the development of interactive robot control using dictionaries, aligning with Learning Task 2. Instruct learners on how to create a dictionary that maps user-friendly text commands to the functions previously defined. Guide them in writing code to receive user input and then execute the corresponding function via dictionary lookup. Emphasise the importance of error handling for invalid commands.

a. **Talk for Learning Strategy**: Use peer code reviews, where learners walk a partner through their command dictionary and logic. Prompt: "Explain how your program responds to the input 'right'."

b. **Experiential Strategy**: Learners run their programs, test multiple command inputs, and modify mappings to simulate expanded robot control scenarios.

c. To support different learning paces, offer a predefined list of commands or a basic dictionary structure as a starting point. Encourage peer collaboration for problem-solving.

4. **Step 4: Assembling a Menu-Driven Robot Interface**

   Direct learners through the process of constructing a comprehensive menu-driven program for a simulated robot, addressing Learning Task 3. Focus on integrating while loops for continuous menu display and conditional statements to manage user selections. Guide them in linking each menu option to a specific function and implementing graceful error messages for incorrect inputs.

   a. **Talk for Learning Strategy**: Before coding, have learners draft their menu logic as a flowchart, then present and justify their design in small groups. Group members ask clarifying questions or suggest improvements.

   b. **Experiential Strategy**: Learners implement and test their interactive menu system, adjusting layout and code structure based on feedback.

   c. For inclusivity, provide a template for the basic menu loop and encourage learners to flowchart their menu logic before coding. Consider offering simpler scenarios for those who benefit from reduced complexity.

5. **Step 5: Consolidating Concepts with a Programming Challenge**

   Present the Prime Number Checker Challenge (Learning Task 4) as an opportunity to consolidate their understanding of functions, conditionals, and loops within a new context. Guide learners in defining the is_prime(n) function, focusing on the logical steps required to determine primality. Encourage systematic testing of their function.

   a. **Talk for Learning Strategy**: After writing the function, learners pair up to explain the logic behind their approach line by line, using prompts like: "What happens when n = 9?" or "Why did you use range(2, int(n**0.5) +1)?"

   b. **Experiential Strategy**: Learners test their function using various inputs and simulate its application in robotic sorting. Optional: Extend the task to a batch-processing scenario using lists and loops.

   c. For differentiation, provide hints on the algorithm's logic for learners who may find the mathematical aspect challenging. Offer a variety of testing scenarios for all learners to ensure comprehensive validation of their code.

6. **Step 6: Program Testing, Debugging, and Reflection**

   Conclude by emphasising the critical importance of testing and debugging in software development. Guide learners through common issues, such as indentation errors or incorrect logic, and provide strategies for identification and resolution. Encourage learners to systematically test their menu program and prime number checker. Promote a culture of peer review where learners can provide constructive feedback on each other's code. Facilitate a reflective discussion on the benefits of modular programming and the challenges encountered, fostering a growth mindset towards programming complexities.

   a. **Talk for Learning Strategy**: Facilitate a final class discussion or gallery walk, where each pair shares one insight from debugging their partner's code or a lesson learnt from the testing phase.

   b. **Experiential Strategy**: Learners revise their programs based on feedback and prepare to use similar modular logic in upcoming robotic simulations.

KEY ASSESSMENT: PRACTICE PAPER 1

DoK Level 3: Short Answer/Justification Questions - Suggested Modes: Home Work/ Individual Presentation

Instructions: Provide comprehensive answers to the following questions, offering detailed justifications, clear explanations, or insightful design considerations as required.

1. **Justifying Modular Design in Large-Scale Robotics [6 Marks]** You are a lead developer on a team building the control software for a next-generation, reconfigurable industrial robot arm. This arm will be deployed in various manufacturing settings, adapting to tasks such as precision welding, automated painting, and intricate assembly operations. Justify, in detail, why structuring this robot's control software with a profound emphasis on using separate, well-defined functions for each distinct operational behaviour (e.g., execute_precision_weld(), perform_tool_change(), Maps_to_waypoint()) represents a superior architectural decision compared to developing a single, monolithic script. Discuss at least three distinct benefits beyond the basic

2. concept of code reuse, focusing on aspects that are critically important for the development and long-term viability of large-scale, adaptable robotics projects.

Hint

- *Conduct **Practice Paper 1 on Week 12**. It should cover contents taught from Year 1 to 3. Refer to the **Table of Specification and Structure of the Paper** at the end of this section. Learners' papers should be marked immediately and scores entered into the SAP.*

Note

- For **Weeks 11 to 16**, select the suggested assessment strategies under the various Focal Areas to conduct the PLCs for those weeks. For **Week 12**, organise **Practice Paper 1.**

Conclusion

In this session, learners build modularity into their programs using functions, a core principle in scalable and reusable code. The creation of custom functions, the use of parameters, return values, and dictionaries support the design of dynamic robot control systems. The inclusion of menus and user input simulations fosters skills in interaction-driven programming.

TABLE OF SPECIFICATION

MULTIPLE CHOICE

Substrand	Indicators	Type	1	2	3	4	Total
Introduction to Physics	**1.1.1.LI.1** Identify careers that are related to physics in various sectors of the economy.	MCQs	1				1
		Essay					
		Practical					
	1.1.1.LI.2 Use basic mathematical concepts to solve problems – trigonometric ratios, Pythagoras's theorem, sine and cosine rule, indices.	MCQs					
		Essay			1		1
		Practical					
	1.1.1.LI.1 Identify the basic units in physics	MCQs	2				2
		Essay					
		Practical					
	1.1.1.LI.2 Determine the dimensions of physical quantities (velocity, acceleration, mass, length, time, weight, energy, force)	MCQs					
		Essay			1		1
		Practical					
	1.1.1.LI.3 Identify the errors in the use of meter rule, protractor, electronic balance, vernier calliper and the micrometer screw gauge.	MCQs					
		Essay					
		Practical					
	1.1.1.LI.4 Explain the types of errors – systematic, random and parallax	MCQs	1				1
		Essay					
		Practical					
	1.1.1.LI.5 Explain scientific notation and their unit multipliers.	MCQs		1			1
		Essay					
		Practical					
	1.1.1.LI.6 Distinguish scalars from vectors (qualitative treatment)	MCQs		2			2
		Essay					
		Practical					

Basic Physics	**2.1.1.LI.1** Explain density	MCQs					
		Essay					
		Practical					
	2.1.1.LI.2 State Archimedes' principle	MCQs		1			1
		Essay					
		Practical					
	2.1.1.LI.3 State principle of flotation	MCQs					
		Essay					
		Practical					
	2.1.1.LI.4 Apply Archimedes' principle and principle of flotation to solve problems	MCQs					
		Essay					
		Practical					
	2.1.1.LI.1 Check validity of equations	MCQs					
		Essay					
		Practical					
	2.1.1.LI.2 Establish the relationship between quantities	MCQs					
		Essay					
		Practical					
	2.1.1.LI.1 Resolve vectors into their components	MCQs					
		Essay					
		Practical					
	2.1.1.LI.2 Determine the resultant of two vectors	MCQs					
		Essay					
		Practical					
	3.1.1.LI.1 Explain moments	MCQs		1			1
		Essay					
		Practical					
	3.1.1.LI.2 State the principle of moments	MCQs					
		Essay					
		Practical					

	3.1.1.LI.3 State the conditions necessary for a body to be in equilibrium	MCQs	2				2
		Essay					
		Practical					
	3.1.1.LI.4 Calculate the reactions at supports	MCQs					
		Essay			1		1
		Practical					
	3.1.1.LI.1 State Newton's law of universal gravitation	MCQs		1			1
		Essay					
		Practical					
	3.1.1.LI.2 Deduce the relationship between acceleration due to gravity, g and universal gravitation constant, G	MCQs					
		Essay					
		Practical					
	3.1.1.LI.3 Deduce the period of rotation of a satellite moving in circular orbit	MCQs					
		Essay					
		Practical					
	3.1.1.LI.4 Explain weightlessness	MCQs					
		Essay					
		Practical					
	3.1.1.LI.1 Explain the universe and its composition as applied to astronomy and cosmology	MCQs		1			1
		Essay					
		Practical					
	3.1.1.LI.2 Explain how our indigenous culture benefits from and applies to astronomy and cosmology.	MCQs					
		Essay					
		Practical					
	3.1.1.LI.3 Explain the formation of the sun, planets and stars	MCQs					
		Essay					
		Practical					
	3.1.1.LI.4 Describe galaxies and their various types – elliptical, spiral, irregular, quasars, blazers, Lenticular, active, seyfert.	MCQs					
		Essay					
		Practical					

Matter	**1.1.2.LI.1** Identify the various states of matter	MCQs	1	1			2
		Essay					
		Practical					
	1.1.2.LI.2 Distinguish between molecular arrangement of the various states	MCQs					
		Essay					
		Practical					
	2.1.2.LI.1 Distinguish elastic deformation from plastic deformation	MCQs		1			1
		Essay			1		1
		Practical					
	2.1.2.LI.2 State Hooke's law	MCQs					
		Essay					
		Practical					
	2.1.2.LI.3 Calculate the energy stored in in an elastic material	MCQs					
		Essay					
		Practical					
	2.1.2.LI.4 Calculate Young's modulus	MCQs					
		Essay					
		Practical					
Kinematics	**1.1.3.LI.1** Describe the various types of motion i.e. circular, oscillatory, rectilinear, spin and random	MCQs		1			1
		Essay					
		Practical					
	1.1.3.LI.2 Establish equations of uniformly accelerated motion and its application in everyday life	MCQs					
		Essay					
		Practical					
	1.1.3.LI.3	MCQs					
	Represent the motion of objects graphically i.e. distance-time, displacement-time and velocity-time as well as deducing equations that can be made from it	Essay					
		Practical					
	2.1.3.LI.1 Identify some games and sports where projectiles are seen of circular motion	MCQs	1				1
		Essay	1				1
		Practical					

	2.1.3.LI.2 Determine the time taken by a projectile to reach maximum height and the maximum height	MCQs					
		Essay					
		Practical					
	2.1.3.LI.3 Determine the range of a projectile	MCQs					
		Essay					
		Practical					
	2.1.3.LI.1 Explain circular motion and give some examples of circular motion	MCQs					
		Essay					
		Practical					
	2.1.3.LI.2 Explain and calculate centripetal force	MCQs		1			1
		Essay					
		Practical					
	2.1.3.LI.3 Explain banking and skidding	MCQs					
		Essay					
		Practical					
	2.1.3.LI.4 Calculate maximum and minimum tension in a string	MCQs					
		Essay					
		Practical					
	3.1.2.LI.1 Explain simple harmonic motion and give some examples of such type of motion	MCQs	1				1
		Essay					
		Practical			1	1	2
	3.1.2.LI.2 Determine the velocity, acceleration and kinetic energy of a body performing SHM	MCQs					
		Essay					
		Practical					
	3.1.2.LI.3 Explain the variation of potential and kinetic energy of a body performing SHM	MCQs					
		Essay					
		Practical					
	3.1.2.LI.4 Determine the energy stored in a system performing SHM	MCQs					
		Essay					
		Practical					
Dynamics	**1.1.4.LI.1** State Newton's laws of motion	MCQs					
		Essay					

	1.1.4.LI.2 Apply Newton's second law to establish the relationship between force, mass and acceleration	MCQs		1			1
		Essay					
		Practical					
	1.1.4.LI.3 Identify daily applications of Newton's laws of motion	MCQs					
		Essay					
		Practical					
	1.1.4.LI.1 Explain how pressure changes with depth in a fluid.	MCQs					
		Essay					
		Practical					
	1.1.4.LI.1 Explain the operation of brake systems in vehicles and the operation of the hydraulic press	MCQs					
		Essay					
		Practical					
	1.1.4.LI.1 State Pascal's principle.	MCQs	2				2
		Essay	1				1
		Practical					
	2.1.4.LI.1 Identify some effects and applications of friction	MCQs					
		Essay		1			1
		Practical					
	2.1.4.LI.2 Determine the coefficient of friction	MCQs					
		Essay					
		Practical					
	3.1.3.LI.1 Distinguish between elastic and inelastic collisions	MCQs					
		Essay					
		Practical					
	3.1.3.LI.2 Use momentum change to establish impulse and hence deduce Newton's second law of motion	MCQs					
		Essay					
		Practical					
	3.1.3.LI.3 Verify Newton's third law of motion from momentum change	MCQs					
		Essay					
		Practical					

Heat	**1.2.1.LI.1**	MCQs					
	Discuss the thermometric substances and properties and their associated characteristics	Essay					
		Practical					
	1.2.1.LI.2	MCQs					
	Describe the features and uses of different types of thermometers	Essay					
		Practical					
	1.2.1.LI.3	MCQs					
	Describe the various temperature scales and the construction of their corresponding thermometers	Essay		1			1
		Practical					
	1.2.1.LI.4	MCQs					
	Derive the relationship between the Celsius, Fahrenheit and the Kelvin scales.	Essay					
		Practical					
	2.2.1.LI.1	MCQs					
	Explain the specific heat capacity and classify substances into good and bad conductors of heat	Essay					
		Practical					
	2.2.1.LI.2	MCQs					
	Calculate the specific heat capacities of substances	Essay					
		Practical					
	2.2.1.LI.3	MCQs					
	Describe how to determine the specific heat capacity of a solid using a liquid of a known specific heat capacity	Essay					
		Practical					
	2.2.1.LI.4	MCQs					
	Explain latent heat of fusion and vaporisation and specific latent heat of fusion and vaporisation	Essay					
		Practical					
	3.2.1.LI.1	MCQs					
	Explain linear, area and volume expansivities	Essay					
		Practical					
	3.2.1.LI.2	MCQs					
	Calculate the linear, area and volume expansivities of metals	Essay			1		1
		Practical					

	3.2.1.LI.3 Group metals in increasing order of expansion property using their linear expansivities	MCQs					
		Essay					
		Practical					
Wave	**1.2.2.LI.1** Describe the process involved in image formation in plane mirrors and their characteristics	MCQs	2				
		Essay	1				1
		Practical					
	1.2.2.LI.2 Deduce the laws of reflection	MCQs					
		Essay					
		Practical					
	1.2.2.LI.3 Determine the number of images formed by inclined mirrors	MCQs					
		Essay					
		Practical					
	1.2.2.LI.1 Explain terminologies associated with spherical mirrors	MCQs					
		Essay					
		Practical					
	1.2.2.LI.2 Describe the processes involved in image formation in spherical mirrors and their characteristics using ray tracing	MCQs					
		Essay					
		Practical		1		1	2
	1.2.2.LI.3 Determine the position and characteristics of images formed by spherical mirrors with mirror formula and magnification formula	MCQs					
		Essay			1		1
		Practical					
	1.2.2.LI.1 Explain refraction and state the laws of refraction	MCQs					
		Essay					
		Practical					
	1.2.2.LI.2 Determine refractive index of a medium	MCQs					
		Essay					
		Practical					
	1.2.2.LI.3 Explain total internal reflection	MCQs					
		Essay					
		Practical					

	1.2.2.LI.4 Establish the relationship between the real depth, apparent depth and the refractive index.	MCQs					
		Essay					
		Practical					
	2.2.2.LI.1 Identify types of waves with properties	MCQs					
		Essay					
		Practical					
	2.2.2.LI.2 Classify waves under longitudinal, transverse, electromagnetic, mechanical, progressive and stationary	MCQs					
		Essay		1			1
		Practical					
	2.2.2.LI.3 Determine the amplitude, velocity, frequency, period, wavelength from a sinusoidal wave signal	MCQs					
		Essay					
		Practical					
	2.2.2.LI.4 Given the amplitude, velocity, frequency, period, wavelength of a wave, deduce the wave equation	MCQs					
		Essay					
		Practical					
	2.2.2.LI.1 Explain the production, nature and transmission of sound	MCQs					
		Essay					
		Practical					
	2.2.2.LI.2 Classify sound into infrasonic, audio-sonic and ultrasonic	MCQs	2				2
		Essay	1				1
		Practical					
	2.2.2.LI.3 Explain how an echo comes about and apply the phenomenon in the determination of the depth of rivers.	MCQs					
		Essay					
		Practical					
	2.2.2.LI.4 Apply the concept of resonance in the determination of speed of sound in air.	MCQs					
		Essay					
		Practical					

	3.2.2.LI.1 Identify the types and features of lenses	MCQs					
		Essay					
		Practical					
	3.2.2.LI.2 Describe how images are formed by lenses	MCQs					
		Essay					
		Practical					
	3.2.2.LI.3 Calculate the image distance and magnification using the lens formula	MCQs		1			1
		Essay					
		Practical					
	3.2.2.LI.1 Identify features of laser configuration	MCQs					
		Essay					
		Practical					
	3.2.2.LI.2 Describe the production of laser beam	MCQs					
		Essay					
		Practical					
	3.2.2.LI.3 Identify the properties of lasers	MCQs	1				1
		Essay					
Electrostatics	**1.3.1.LI.1** Explain how the gold leaf electroscope can be used to detect the charge carried by a body	MCQs					
		Essay					
		Practical					
	1.3.1.LI.2 Identify electrons as mobile charge carriers	MCQs					
		Essay					
		Practical					
	1.3.1.LI.3 Explain how charge carriers in conductors, semiconductors and insulators behave.	MCQs					
		Essay					
		Practical					
	1.3.1.LI.4 Explain the distribution of charges on surfaces; spherical, pear-shaped and sharp point	MCQs					
		Essay					
		Practical					

	1.3.1.LI.1 Define charge as a fundamental property of matter(like mass)	MCQs	1				1
		Essay					
		Practical					
	1.3.1.LI.2 Explain the conservation of charge and its behaviour	MCQs					
		Essay					
		Practical					
	1.3.1.LI.3 Differentiate between two charges (positive and negative)	MCQs					
		Practical					
	2.3.1.LI.1 Explain and show how an electric field is represented and their characteristics	MCQs					
		Essay					
		Practical					
	2.3.1.LI.2 State Coulomb's law of electrostatics and calculate electrostatic force acting on the charges	MCQs					
		Essay					
		Practical			1		1
	2.3.1.LI.3 Determine the electric field/ intensity at a point due to a point charge	MCQs		1			1
		Essay					
		Practical					
	1.3.1.LI.4 Define and calculate potential difference	MCQs					
		Essay					
		Practical					
	2.3.1.LI.1 Describe the structure and operation of capacitors	MCQs					
		Essay					
		Practical					
	2.3.1.LI.2 Determine the effective capacitance of a number of capacitors arranged in : (i) series (ii) parallel	MCQs					
		Essay			1		1
		Practical					
	2.3.1.LI.3 Determine the capacitance of a parallel plate capacitors	MCQs					
		Essay					
		Practical					

	2.3.1.LI.4 Describe the behaviour of a capacitor in DC and AC circuits	MCQs					
		Essay					
		Practical					
	2.3.1.LI.5 Calculate the energy stored in a capacitor	MCQs					
		Essay					
		Practical					
Direct current	**3.3.1.LI.1** Explain the key terminologies of circuitry and how they are related to one another	MCQs					
		Essay					
		Practical					
	3.3.1.LI.2 Express Ohm's law and its relationship to electrical resistance; ohmic and non-ohmic conductors	MCQs		1			1
		Essay			1		1
		Practical				1	1
	3.3.1.LI.1 Describe and demonstrate the structure and working principle of resistors and their arrangements in circuits	MCQs		1			1
		Essay			1		1
		Practical		1			1
	3.3.1.LI.2 Describe and demonstrate the types of variable resistors i.e. potentiometers and rheostats and their working principles.	MCQs					
		Essay					
		Practical					
	3.3.1.LI.1 State Kirchhoff's laws of electrical networks	MCQs		1			1
		Essay					
		Practical					
	3.3.1.LI.2 Determine the current flowing through a branch in electrical networks	MCQs					
		Essay					
		Practical					
	3.3.1.LI.3 Discuss how a galvanometer can be adapted as an ammeter and a voltmeter	MCQs					
		Essay					
		Practical					
Magnetostatics	**1.3.2.LI.1** Distinguish between magnets, magnetic and non-magnetic materials.	MCQs	2				2
		Essay	1				1
		Practical					

	1.3.2.LI.2 Describe the magnetic field	MCQs	1				1
		Essay					
		Practical					
	1.3.2.LI.3 Describe the processes involved in magnetisation and demagnetisation	MCQs					
		Essay					
		Practical					
Electromagnetism	**2.3.2.LI.1** Describe the force exerted on a current-carrying conductor in a magnetic field	MCQs					
		Essay					
		Practical					
	2.3.2.LI.2 Discuss the factors that affect the magnitude of the magnetic force on a current-carrying conductor in a magnetic field	MCQs	1				1
		Essay					
		Practical					
	2.3.2.LI.3 Fleming's left-hand rule	MCQs					
		Essay					
		Practical					
	2.3.2.LI.1 Explain the forces acting between parallel conductors carrying current in a magnetic field	MCQs					
		Essay					
		Practical					
	2.3.2.LI.2 Explain the torque on a rectangular current-carrying coil in a magnetic field	MCQs					
		Essay					
		Practical					
	2.3.2.LI.3 Describe and analyse the structure and working principle of the motor and moving coil galvanometer	MCQs					
		Essay					
		Practical					
	2.3.2.LI.1 Calculate force on a charged particle in a magnetic field.	MCQs					
		Essay					
		Practical					
	2.3.2.LI.2 Describe a force on a moving charged particle in an electric field	MCQs					
		Essay					
		Practical					

	2.3.2.LI.2 Calculate a force on a moving charged particle in a crossed field (Lorentz force)	MCQs					
		Essay					
		Practical					
Alternating current	**3.3.2.LI.1** Discuss the characteristics of alternating current/voltage	MCQs					
		Essay					
		Practical					
	3.3.2.LI.2 Distinguish between the types of AC waveforms	MCQs					
		Essay					
		Practical					
	3.3.2.LI.3 Using mathematical relationships to define alternating current and alternating voltage	MCQs					
		Essay					
		Practical					
	3.3.2.LI.1 Describe a.c. circuit connections using resistors, inductors, capacitors in series	MCQs					
		Essay					
		Practical					
	3.3.2.LI.2 Describe and distinguish between resistance and impedance	MCQs					
		Essay					
		Practical					
	3.3.2.LI.3 Describe the power triangle and explain the power factor using mathematical concepts	MCQs					
		Essay					
		Practical					
Analogue Electronics	**1.3.3.LI.1** Describe the formation of the two types of conductors; n-type and p-type	MCQs					
		Essay		1			1
		Practical					
	1.3.3.LI.2 Describe the basic structure and application of the PN junction diode in reverse and forward-biased circuits	MCQs					
		Essay					
		Practical					
	1.3.3.LI.3 Analyse the benefits of using LEDs and I-V characteristics of Zener diodes and applications	MCQs					
		Essay					
		Practical					

	1.3.3.LI.1 Explain the terminologies “input” and “output” of a transducer with examples	MCQs		1			1
		Essay					
		Practical					
	1.3.3.LI.2 Explain the effect of temperature changes on resistance using a thermistor, light dependent resistor (LDR), infra-red diode and the microphone to analyse the characteristic graphs of thermistors and LDRs	MCQs					
		Essay					
		Practical					
	1.3.3.LI.3 Describe the processes involved in the operation of the following transducers; microphone, loudspeaker, buzzer, low voltage DC motor, electromagnetic relays and infra-red diode	MCQs	1				1
		Essay					
		Practical					
	1.3.3.LI.1 Describe the construction and action of the bipolar junction transistor	MCQs					
		Essay					
		Practical					
	1.3.3.LI.2 Describe transistor biasing	MCQs					
		Essay					
		Practical					
	1.3.3.LI.3 Describe various transistor configurations and the use of an N-P-N transistor as a small signal amplifier.	MCQs					
		Essay					
		Practical					
Digital Electronics	**2.3.3.LI.1** Describe analogue and digital signals	MCQs	1				1
		Essay					
		Practical					
	2.3.3.LI.2 Distinguish between pull-up and pull-down resistors	MCQs					
		Essay					
		Practical					

	2.3.3.LI.3 Describe and use the 7-segment display module	MCQs					
		Essay					
		Practical					
	2.3.3.LI.1 Distinguish between analogue and digital signals and describe their functions	MCQs	2				2
		Essay	1				1
		Practical					
	2.3.3.LI.2 Describe the characteristics of basic logic and universal gates	MCQs					
		Essay					
		Practical					
	2.3.3.LI.3 Describe and explain sum of product (Boolean expression)	MCQs					
		Essay					
		Practical					
	2.3.3.LI.4 Describe combinational logic applications and microcontroller.	MCQs					
		Essay					
		Practical					
	2.3.3.LI.1 Describe simple integrated circuits	MCQs					
		Essay					
		Practical					
	2.3.3.LI.2 Design and describe the fabrication of a simple integrated circuit	MCQs					
		Essay					
		Practical					
Electromagnetic induction and application	**3.3.3.LI.1** Explain electromagnetic induction	MCQs					
		Essay					
		Practical					
	3.3.3.LI.2 Using experimentation, verify the laws of electromagnetic induction	MCQs					
		Essay					
		Practical					
	3.3.3.LI.3 Describe the factors that affect induced emf and explain direction of induced current using mnemonics and Fleming's Right Hand Rule	MCQs					
		Essay					
		Practical					

	3.3.3.LI.1 Explain inductance and obtain a mathematical expression for the relationship	MCQs					
		Essay					
		Practical					
	3.3.3.LI.2 Discuss the behaviour of the inductor in a.c and d.c. circuits	MCQs					
		Essay					
		Practical					
	3.3.3.LI.3 Explain how energy is stored in an inductor.	MCQs					
		Essay					
		Practical					
	3.3.3.LI.1 Discuss the structure, uses and principle of operation of a transformer	MCQs					
		Essay					
		Practical					
	3.3.3.LI.2 Discuss the factors that affect the transmission of power and efficiency of a transformer and ways of improving its efficiency	MCQs					
		Essay					
		Practical					
	3.3.3.LI.3 Explain Eddy current, effects and applications	MCQs					
		Essay					
		Practical					
Application of electronics	**3.3.4.LI.1** Design a half wave rectifier	MCQs					
		Essay					
		Practical					
	3.3.4.LI.2 Design a full wave rectifier	MCQs					
		Essay					
		Practical					
	3.3.4.LI.3 Design a smoother/filter circuit using a capacitor	MCQs					
		Essay					
		Practical					
	3.3.4.LI.1 Design a PNP amplifier	MCQs					
		Essay					
		Practical					

	3.3.4.LI.2 Design an NPN amplifier	MCQs					
		Essay					
		Practical					
	3.3.4.LI.1 Design a darkness detector	MCQs					
		Essay					
		Practical					
	3.3.4.LI.2 Design a battery charger	MCQs					
		Essay					
		Practical					
Atomic Physics	**1.4.1.LI.1** Explain various atomic models and their limitations	MCQs	1				1
		Essay					
		Practical					
	1.4.1.LI.2 Calculate the energy of a photon during a transition	MCQs		1			1
		Essay					
		Practical					
	2.4.1.LI.1 Explain photoelectric effect and wave-particle duality	MCQs	1				1
		Essay					
		Practical					
	2.4.1.LI.2 State the laws of photoelectric effect	MCQs					
		Essay					
		Practical					
	2.4.1.LI.3 Identify areas of application of photoelectric effect and how they are applied	MCQs					
		Essay					
		Practical					
	3.4.1.LI.1 Describe how x-rays are produced	MCQs	1				1
		Essay		1			1
		Practical					
	3.4.1.LI.2 Distinguish between hard x-rays and soft x-rays	MCQs					
		Essay					
		Practical					
	3.4.1.LI.3 Calculate the energy, frequency and wavelength of x-rays	MCQs					
		Essay					
		Practical					

	3.4.1.LI.4 Identify sectors where x-rays are applied and how they are applied	MCQs					
		Essay					
		Practical					
Nuclear Physics	**1.4.2.LI.1** Describe the structure of the nucleus of the atom	MCQs		1			1
		Essay					
		Practical					
	1.4.2.LI.2 Explain radioactivity	MCQs					
		Essay	1				1
		Practical					
	1.4.2.LI.3 Balanced basic nuclear reactions	MCQs	1				1
		Essay					
		Practical					
	2.4.2.LI.1 State the radioactive decay law	MCQs					
		Essay					
		Practical					
	2.4.2.LI.2 Calculate the half-life of a radioactive sample	MCQs					
		Essay			1		1
		Practical					
	2.4.2.LI.3 Calculate the age of a bone specimen	MCQs					
		Essay					
		Practical					
	3.4.2.LI.1 Calculate the energy released in a nuclear reaction	MCQs		1			1
		Essay		1			1
		Practical					
	3.4.2.LI.2 Distinguish between fission and fusion	MCQs					
		Essay		1			1
		Practical					
	3.4.2.LI.3 Describe the process of power generation using a nuclear reactor	MCQs					
		Essay					
		Practical					
	Total		**36**	**30**	**12**	**3**	**81**

PLC Session 12

45 minutes	Planning & Reviewing the Learning Plan

1. Begin the session by recapping lessons from the previous session, focusing on how learning was applied.
2. Use your subject specific app to explore the content before planning. Ask the app to
   a. explain the week's concepts in a practical way that will make it easier for learners to understand.
   b. identify three common misconceptions learners may have on the week's content and how to address them using contexts familiar to Ghanaian learners (NTS 3a, 3g and 3m).
3. Use your subject specific app to plan your lesson. Ask the app to
   a. suggest the most effective method for teaching the week's concepts to learners with diverse backgrounds.
   b. suggest learning activities, differentiation and essential questions based on the method of teaching (NTS 3a-3q).
   c. review your prepared learning plan and address any challenging issues you may have identified in your subject area (NTS 3a-3c).

Note

- Take note of the recap topics linked to Years 1 and 2 while preparing for your lesson.
- Refer to the Values Learning Community Handbook and link the values for the week in your lesson to ensure that learners are applying what they learn.

4. Use your app to plan and administer the mode of assessment for the Student Assessment Portal (SAP).
   - The key assessment for the week is ***Practice Paper 1***. Chat with your app to plan this assessment by reviewing or developing the assessment items/tasks and marking schemes/rubrics for the practice paper (NTS 3k-3q).

10 minutes	Content Exploration

5. As a subject group, select one problem or topic/idea/term in the week's content that is challenging (not a routine procedural question, but one that requires conceptual understanding) and then
   a. use the app, ask it to solve the problem step by step and explain the reasoning behind each step (you may be assisted by a teacher who is strong in the content area)
   b. all teachers in the group work through the problem individually for 3–4 minutes

  c. the group then works through it together the problem, step by step

  d. highlight the most common errors made at each step (NTS 1c, 1e, 2c, 2e, 3e-3j).

Note

- This part is not 'How do I teach this?' but how do I teach it for learners to understand?
- If the group still shares the same content gap, use the app: ask it to solve the problem step by step and explain the reasoning behind each step.
- Problems/ideas discussed should be sourced from the Teacher Manual content, not invented separately. This keeps the content exploration directly connected to what teachers will deliver.

30 minutes	Enactment

6. Enact a teaching activity based on your learning plan (NTS 1a-1b, 1e-1g, 2c, 2e, 2f, 3c-3j and 3o).

  a. Colleagues observe with specific attention to clarity of concept explanation, use of familiar Ghanaian examples and how misconceptions are handled if they arise.

  b. After the enactment, ask for structured feedback from the group

    i. was the concept explained in a way that a learner with no prior knowledge could follow?

    ii. were the examples grounded in contexts familiar to Ghanaian learners?

    iii. did any misconceptions surface and if so, how were they addressed?

5 minutes	Reflection

7. Reflect and share your views on the session. One thing that worked well, what did not work well and what you will do differently. (NTS 1a, 1b).

8. Identify a critical friend to observe and provide feedback on your lesson in relation to the PLC Session conducted (NTS 1f, 1g).

9. Remember to

  a. transfer your chats into your learning planner template.

  b. read your Teacher Manual and related Learner Material in preparation for the next session. Take note of the recap weeks, which are linked to Years 1 and 2 (NTS 3a).

  c. integrate National Values, GESI, SEL, and 21st century skills into your learning activities, assessment tasks, and rubrics/mark schemes.

Hint

- *Conduct **Practice Paper 2 on Week 18**. It should cover the contents taught from Year 1 to 3. Refer to the **Table of Specification and Structure of the Paper** at the end of this section to guide you. Learners' papers should be marked immediately and scores entered into the SAP.*

PLC Session 17 & 18

45 minutes	Planning & Reviewing the Learning Plan

1. Begin the session by recapping lessons from the previous session, focusing on how learning was applied.
2. Use your subject specific app to explore the content before planning. Ask the app to
   a. explain the week's concepts in a practical way that will make it easier for learners to understand.
   b. identify three common misconceptions learners may have on the week's content and how to address them using contexts familiar to Ghanaian learners (NTS 3a, 3g and 3m).
3. Use your subject specific app to plan your lesson. Ask the app to
   a. suggest the most effective method for teaching the week's concepts to learners with diverse backgrounds.
   b. suggest learning activities, differentiation and essential questions based on the method of teaching (NTS 3a-3q).
   c. review your prepared learning plan and address any challenging issues you may have identified in your subject area (NTS 3a-3c).

Note

- Take note of the recap topics linked to Years 1 and 2 while preparing for your lesson.
- Refer to the Values Learning Community Handbook and link the values for the week in your lesson to ensure that learners are applying what they learn.

4. Use your app to plan and administer the mode of assessment for the Student Assessment Portal (SAP).
   - The key assessment for the week is ***Practice Paper 2***. Chat with your app to plan this assessment by reviewing or developing the assessment items/tasks and marking schemes/rubrics for the practice paper (NTS 3k-3q).

10 minutes	Content Exploration

5. As a subject group, select one problem or topic/idea/term in the week's content that is challenging (not a routine procedural question, but one that requires conceptual understanding) and then
   a. use the app, ask it to solve the problem step by step and explain the reasoning behind each step (you may be assisted by a teacher who is strong in the content area)
   b. all teachers in the group work through the problem individually for 3–4 minutes
   c. the group then works through it together the problem, step by step
   d. highlight the most common errors made at each step (NTS 1c, 1e, 2c, 2e, 3e-3j).

Note

1. This part is not 'How do I teach this?' but how do I teach it for learners to understand?
2. If the group still shares the same content gap, use the app: ask it to solve the problem step by step and explain the reasoning behind each step.
3. Problems/ideas discussed should be sourced from the Teacher Manual content, not invented separately. This keeps the content exploration directly connected to what teachers will deliver.

30 minutes	Enactment

6. Enact a teaching activity based on your learning plan (NTS 1a-1b, 1e-1g, 2c, 2e, 2f, 3c-3j and 3o).
   a. Colleagues observe with specific attention to clarity of concept explanation, use of familiar Ghanaian examples and how misconceptions are handled if they arise.
   b. After the enactment, ask for structured feedback from the group
      i. was the concept explained in a way that a learner with no prior knowledge could follow?
      ii. were the examples grounded in contexts familiar to Ghanaian learners?
      iii. did any misconceptions surface and if so, how were they addressed?

5 minutes	Reflection

7. Reflect and share your views on the session. One thing that worked well, what did not work well and what you will do differently. (NTS 1a, 1b).
8. Identify a critical friend to observe and provide feedback on your lesson in relation to the PLC Session conducted (NTS 1f, 1g).

9. Remember to

a. transfer your chats into your learning planner template.

b. read your Teacher Manual and related Learner Material in preparation for the next session. Take note of the recap weeks, which are linked to Years 1 and 2 (NTS 3a).

c. integrate National Values, GESI, SEL, and 21st century skills into your learning activities, assessment tasks, and rubrics/mark schemes.

Section Review

This section presents the foundational shift from block-based programming to structured Python programming for robotics. Emphasis was placed on understanding syntax, variables, and core data types as applied to robot control. Gradually, learners developed the ability to construct logic using conditionals and loops, followed by the introduction of functions to support modular program design. Each concept was embedded in simulated robotics tasks, reinforcing the link between programming structure and real-time robot behaviour. The section concludes with activities that combine all elements into complete programs, supporting autonomous decision-making and efficient robotic operations.

ADDITIONAL READING

Resource	QR Code to Accessing Resource
1. Easy Python Coding for Robots: Getting Started (Video Resource).	
2. Python3 for Robotics – ROS/Robotignite Academy Course (Online Course)	
3. Introduction to Programming with Python for Middle and High School Students (Online Course)	

MARKING SCHEMES FOR THE KEY ASSESSMENTS

WEEKS 11 – 16

RUBRIC - SHORT ANSWER AND EXTENDED RESPONSE ESSAY

Criteria	Mark Distribution	
Justification for Python	**2 Marks**	Provides two distinct, well-reasoned justifications directly linking Python's capabilities (scalability, control, integration, debugging) to the humanitarian robot's complex needs.
Linking to Python's capabilities	**2 Marks**	Links Python's capabilities (scalability, control, integration, debugging) to the humanitarian robot's complex needs.
Data Strategy & Justification (10 Marks)	**5 Marks**	Correctly identifies appropriate data types for all five pieces of information
	5 Marks	provides clear, logical justifications for each choice based on data characteristics.
Advantage of Lists	**6 Marks**	Explains with at least 3 points on how lists provide superior advantages (e.g., ordered collection, dynamic size, easy iteration, grouped access) for managing historical events compared to individual variables.
Total	**20 Marks**	

Individual Practical Task (*18 Marks*) Add to this, 2 bonus marks for answering b (iv) = grand total 20

CRITERIA	MARKS	DESCRIPTION
Data Generation and Storage (temperature readings list)	**2 marks**	At least 10 floating-point values provided in a properly initialised Python list.
	1 mark	Less than 10 values or mostly integers used, or list incorrectly initialised (e.g., missing brackets).
Data Generation and Storage (humidity readings list)	**2 marks**	At least 10 integer values correctly stored in a Python list, corresponding to temperature readings.
	1 mark	Fewer than 10 values, or values are not integers, or the list is loosely defined.
Data Generation and Storage (robot path list of coordinate tuples)	**2 marks**	List of at least 5 valid tuples with distinct (x, y) coordinates, structured correctly (e.g., [(0,0), (1,2), (2,3)]).
	1 mark	List created with fewer than 5 tuples or syntax issues (e.g., missing parentheses).

Logical alignment & syntax	**2 marks**	All three lists follow logical structure, naming conventions, and valid Python syntax.
	1 mark	Minor syntax errors or inconsistencies (e.g., variable mismatch) that affect readability but not logic.
Data Analysis & Calculation (Average temperature calculation)	**2 marks**	Correct use of Python (e.g., sum()/len()) to compute average; accurate result.
	1 mark	Attempt made, but wrong formula or slight logic error (e.g., sum used without len).
Data Analysis & Calculation (Maximum humidity)	**1 mark**	Correct use of max() or equivalent logic to extract highest value from humidity readings.
Data Analysis & Calculation (Total distance travelled (step-based method))	**2 marks**	Correct count of steps = len(robot_path) - 1, with clear output.
	1 mark	Step count attempted but miscalculated or explained poorly.
Structured Output (Display of all original lists)	**1 mark**	Lists (temperature_readings, humidity_readings, robot_path) displayed clearly in the output.
Structured Output (Display of all calculated values)	**2 marks**	All three values (average temperature, max humidity, total distance) displayed with appropriate labels.
	1 mark	Two of the three displayed or labels are confusing.
Structured Output (Report formatting and clarity)	**2 marks**	Report includes clear headings, proper line spacing, consistent variable names, and user-friendly layout.
	1 mark	Some effort made, but formatting is inconsistent or cluttered.

1. Code Modification Proposal (*10 Marks*)

Score	Description of Performance
10	Fully implements all requirements with accurate and syntactically correct Python code. Uses correct and clearly structured control flow (e.g., for loop with break, if-elif-else) for all parts of the search and termination logic. Comments are clear and relevant.
9	Implements all major requirements accurately. Minor errors in formatting or code style, but logic and syntax are correct. Control structures are appropriate and well-placed.
8	All core elements are present, but with minor logic issues (e.g., missing a break condition, minor misplacement of loop). Code still functions as intended with small corrections.
7	Most requirements implemented. May have logic errors in handling retries or line detection but demonstrates understanding of control flow. Code requires moderate revision to work.
6	Code attempts full implementation but misuses some control structures (e.g., uses while where for is clearer). Code is not reliably functional.
5	Partial code that includes only parts of the solution (e.g., attempts left-turn and right-turn but no retry logic or termination). Basic syntax is correct but incomplete.

4	Limited attempt. Some idea of solution shown but poorly implemented; major gaps in logic and structure.
3	Minimal attempt. Fragmented or incorrect code; shows very limited understanding of control flow.
2	Very weak attempt. Isolated code snippets with little or no relevance to the task.
1	Attempt made but completely incorrect or incoherent. No meaningful alignment with requirements
0	No attempt or response is entirely irrelevant.

2. Justification of Control Structures (*10 Marks*)

Score	Description of Performance
10	Clearly and accurately explains the choice of each control structure (e.g., why for loop for retries, break for early exit). Justifications are logical, concise, and show deep understanding of program flow and robotics.
9	Excellent explanation of most control structures. Slightly vague or missing explanation for one element (e.g., forgets to justify continue, but others are well explained).
8	Justifies most structures well. Some descriptions are generic but understandable (e.g., "for loop is good for counting").
7	Provides general reasoning behind structures but lacks clarity or precision. May misattribute purpose (e.g., "used while because it repeats" without explaining why that matters).
6	Some attempt to explain logic but uses incorrect or confusing reasons. Partial understanding of how control structures affect behaviour.
5	Gives vague reasoning or explains only one part (e.g., only justifies the use of if-else). No reference to loops or flow termination.
4	Very limited explanation, mostly speculative or unclear.
3	Mentions structure names only, with no valid explanation.
2	Very weak attempt. Explanations are largely incorrect or irrelevant.
1	Attempt made but shows no meaningful understanding of control structures.
0	No attempt or response is entirely irrelevant.

3. Testing Strategy (*5 Marks*)

Score	Description of Performance
5	Detailed and complete plan covering all scenarios: (1) brief off-line correction, (2) 3 retries with no success, and (3) final report and halt. Includes explanation of how to simulate or test each case.
4	Covers all three scenarios with clarity but lacks depth in how to carry out the tests or misses small practical details.
3	Covers only two out of three scenarios OR gives minimal detail in how tests would be executed.
2	A very basic outline of testing with unclear steps or only one case considered.

1	Mentions testing vaguely without addressing key behaviour.
0	No attempt or response is entirely irrelevant.

Marking Scheme (6 *Marks*)

1. ***2 marks***: Provides a clear and precise explanation of the fundamental superiority of modular design for large-scale, adaptable robotics.
2. ***1 mark per distinct benefit (maximum 3 marks):*** Each identified benefit (e.g., enhanced maintainability, facilitation of parallel development among sub-teams, improved fault isolation/streamlined troubleshooting, increased scalability for adding future functionalities, simplified testing of individual components) is clearly articulated and relevant to the context.
3. ***1 mark:*** The overall argument is logically coherent, well-structured, and presented with formal language.
4. ***Evaluating Robust Error Handling Strategies in Critical Robotics [8 Marks]:*** In a mission-critical autonomous drone navigation system, flight path coordinates and operational commands are frequently entered by human operators via a text-based terminal. The system's reliability hinges on robust handling of invalid user input (e.g., non-numeric entries for coordinates, values falling outside safe operational ranges). Compare and contrast two distinct approaches for managing such invalid inputs within a menu-driven interface:

   a. ***Approach A:*** A basic else block positioned at the conclusion of an if-elif chain, which simply prints a generic "Invalid Input" message.

   b. ***Approach B:*** A dedicated, reusable error-handling function, process_input_validation_error(input_value, expected_format), which, upon detecting an error, logs the invalid input for debugging, provides specific and actionable feedback to the operator (e.g., "Error: Please enter a numeric value between 0 and 100."), and offers clear options to retry the input or return to the main menu. Evaluate which of these two approaches is demonstrably superior for a mission-critical robotics application. Justify your reasoning by thoroughly discussing the implications of each approach on the system's overall reliability, the operator's user experience, and the efficiency of system debugging in a high-stakes environment.

Marking Scheme (8 *Marks*)

- ***2 marks:*** Presents a clear and accurate comparison of the fundamental functionality of Approach A versus Approach B.
- ***2 marks:*** Provides a detailed evaluation of Approach A's implications, highlighting its limitations regarding user feedback, debugging difficulty, and impact on system reliability in critical contexts.
- ***2 marks***: Provides a detailed evaluation of Approach B's implications, explaining how it significantly enhances user guidance, improves system reliability through robust handling, and streamlines debugging via structured logging.

- ***2 marks*:** Offers a compelling and well-reasoned justification for why Approach B is unequivocally superior for mission-critical applications, synthesising the comparative implications effectively.

4. ***Designing Functional Flow for a Dynamic Sensor-Actuator System [10 Marks*]** You are tasked with designing the core software logic for a smart agricultural robot responsible for maintaining optimal soil conditions in a crop field. This robot must continuously monitor both soil moisture levels and ambient light intensity using its onboard sensors. If the soil moisture level drops below a predefined critical threshold OR the ambient light intensity exceeds a scorching threshold, the robot must immediately activate a specialised watering system. Following the activation and completion of watering, the robot is required to re-verify if the environmental conditions have effectively improved. Use either detailed pseudocode or a comprehensive block diagram to illustrate how you would effectively structure the functions, their parameters, their return values, and the overall flow of information to implement this dynamic sensing, decision-making, and control system. Explicitly emphasise how the use of functions enables a clear separation of concerns (e.g., dedicated functions for sensing, decision-making, actuation, and verification) and how data is precisely passed between these distinct functional modules.

Marking Scheme (*10 Marks*)

- ***2 marks*:** Clearly identifies and names at least three distinct, appropriate functions for the specified tasks (e.g., read_environment_sensors(), evaluate_conditions(), activate_irrigation_system(), reverify_environment()).
- ***3 marks*:** Accurately represents the function signatures, including clearly defined parameters and appropriate return values that facilitate the correct flow of data between functions.
- ***3 marks*:** Demonstrates a logical and coherent flow of the main program loop (or sequential execution), illustrating precisely how functions are invoked based on conditional logic and how their return values meaningfully influence subsequent actions.
- ***2 marks*:** Provides a clear explanation (through pseudocode comments or detailed diagram annotations) of how the design achieves a strong separation of concerns and articulates how critical data is systematically passed and utilised between the different functional modules.