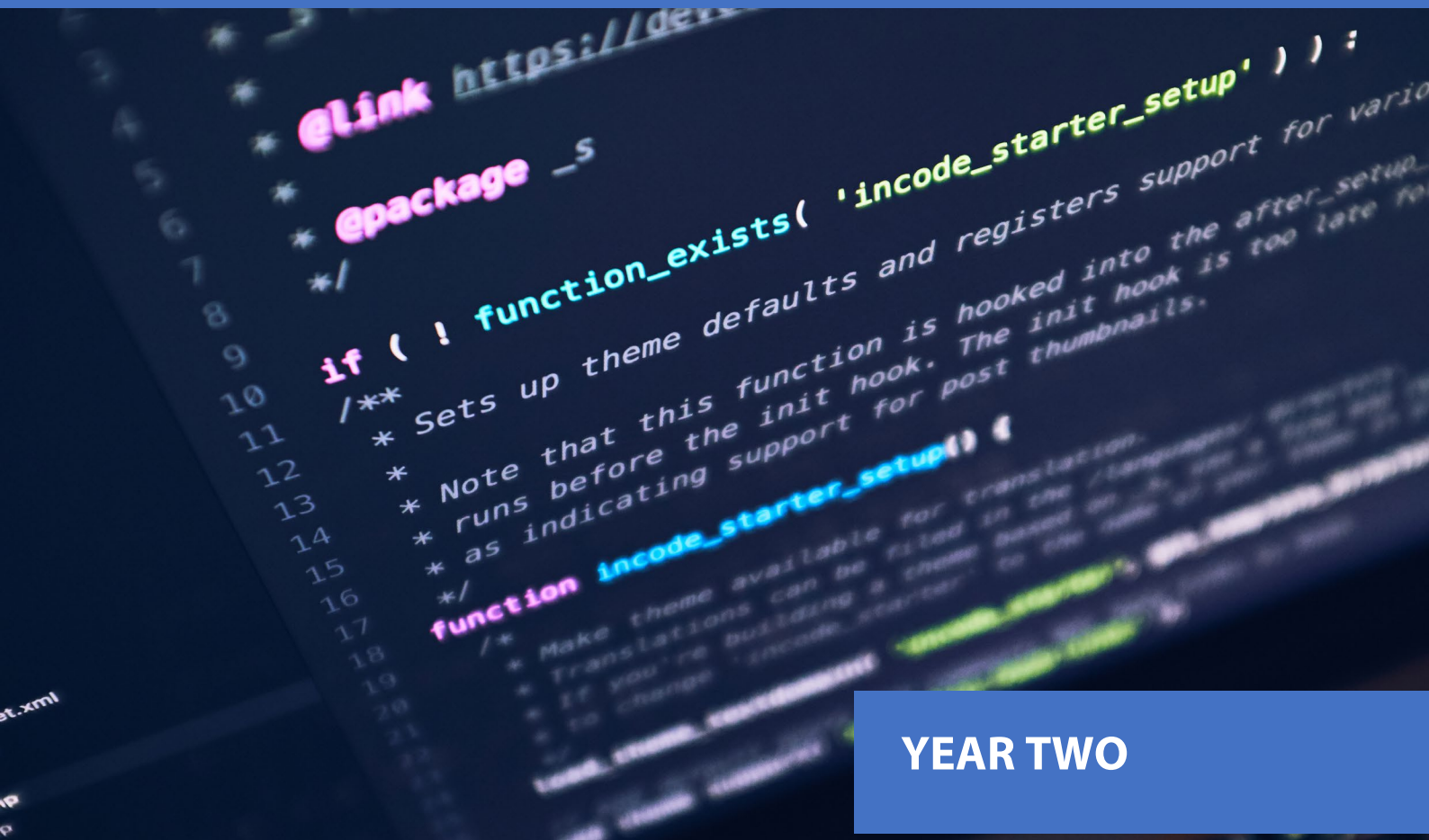
MINISTRY OF EDUCATION

COMPUTING

For Senior High Schools

TEACHER MANUAL



YEAR TWO

NATIONAL COUNCIL FOR CURRICULUM & ASSESSMENT OF MINISTRY OF EDUCATION

MINISTRY OF EDUCATION

REPUBLIC OF GHANA

Computing

For Senior High Schools

Teacher Manual

Year Two

NATIONAL COUNCIL FOR CURRICULUM & ASSESSMENT OF MINISTRY OF EDUCATION

COMPUTING 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

mastercard foundation

T-TEL
Transforming Teaching, Education & Learning

CONTENTS

INTRODUCTION

The National Council for Curriculum and Assessment (NaCCA) has developed a new Senior High School (SHS) curriculum which aims to ensure that all learners achieve their potential by equipping them with 21st Century skills, competencies, character qualities and shared Ghanaian values. This will prepare learners to live a responsible adult life, further their education and enter the world of work.

This is the first time that Ghana has developed an SHS Curriculum which focuses on national values, attempting to educate a generation of Ghanaian youth who are proud of our country and can contribute effectively to its development.

This Teacher Manual for Computing is a single reference document which covers all aspects of the content, pedagogy, teaching and learning resources and assessment required to effectively teach Year Two of the new curriculum. It contains information for all 24 weeks of Year Two including the nine Key Assessments required for the Student Transcript Portal (STP).

Thank you for your continued efforts in teaching our children to become responsible citizens.

It is our belief that, if implemented effectively, this new curriculum will go a long way to transforming our Senior High Schools and developing Ghana so that we become a proud, prosperous and values-driven nation where our people are our greatest national asset.

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	
Name of Staff	**Designation**
Eric Amoah	Deputy Director-General, Technical Services
Reginald Quartey	Ag. Director, Curriculum Development Directorate
Anita Cordei Collison	Ag. Director, Standards, Assessment and Quality Assurance Directorate
Rebecca Abu Gariba	Ag. Director, Corporate Affairs
Anthony Sarpong	Director, Standards, Assessment and Quality Assurance Directorate
Uriah Kofi Otoo	Senior Curriculum Development Officer (Art and Design Foundation & Studio)
Nii Boye Tagoe	Senior Curriculum Development Officer (History)
Juliet Owusu-Ansah	Senior Curriculum Development Officer (Social Studies)
Ayuuba Sullivan Akudago	Senior Curriculum Development Officer (Physical Education & Health)
Godfred Asiedu Mireku	Senior Curriculum Development Officer (Mathematics)
Samuel Owusu Ansah	Senior Curriculum Development Officer (Mathematics)
Thomas Kumah Osei	Senior Curriculum Development Officer (English)
Godwin Mawunyo Kofi Senanu	Assistant Curriculum Development Officer (Economics)
Joachim Kwame Honu	Principal Standards, Assessment and Quality Assurance Officer
Jephtar Adu Mensah	Senior Standards, Assessment and Quality Assurance Officer
Richard Teye	Senior Standards, Assessment and Quality Assurance Officer
Nancy Asieduwaa Gyapong	Assistant Standards, Assessment and Quality Assurance Officer
Francis Agbalenyo	Senior Research, Planning, Monitoring and Evaluation Officer
Abigail Birago Owusu	Senior Research, Planning, Monitoring and Evaluation Officer
Ebenezer Nkuah Ankamah	Senior Research, Planning, Monitoring and Evaluation Officer
Joseph Barwuah	Senior Instructional Resource Officer
Sharon Antwi-Baah	Assistant Instructional Resource Officer
Dennis Adjasi	Instructional Resource Officer

Samuel Amankwa Ogyampo	Corporate Affairs Officer
Seth Nii Nartey	Corporate Affairs Officer
Alice Abbew Donkor	National Service Person

Subject	Writer	Designation/Institution
Additional Mathematics	Dr. Nana Akosua Owusu-Ansah	University of Education Winneba
	Gershon Kwame Mantey	University of Education Winneba
	Innocent Duncan	KNUST Senior High School
Agricultural Science	David Esela Zigah	Achimota School
	Prof. J.V.K. Afun	Kwame Nkrumah University of Science and Technology
	Issah Abubakari	Half Assini Senior High School
	Mrs. Benedicta Carbilba Foli	Retired, Pope John SHS and Minor Seminary
Agriculture	Esther Fobi Donkor	University of Energy and Natural Resources, Sunyani
	Prof. Frederick Adzitey	University for Development Studies
	Eric Morgan Asante	St. Peter's Senior High School
Automotive and Metal Technology	Dr. Sherry Kwabla Amedorme	Akenten Appiah Menka University of Skills Training and Entrepreneurial Development (AAMUSTED)
	Kunkyuuri Philip	Kumasi Senior High Technical School
	Emmanuel Korletey	Benso Senior High Technical School
	Philip Turkson	G E S
Electrical and Electronics Technology	Walter Banuenumah	Akenten Appiah Menka University of Skills Training and Entrepreneurial Development (AAMUSTED)
	Akuffo Twumhene Frederick	Koforidua Senior High Technical School
	Gilbert Second Odjamgba	Ziavi Senior High Technical School
Building Construction and Woodwork Technology	Wisdom Dzidzienyo Adzraku	Akenten Appiah Menka University of Skills Training and Entrepreneurial Development (AAMUSTED)
	Michael Korblah Tsorgali	Akenten Appiah Menka University of Skills Training and Entrepreneurial Development (AAMUSTED)
	Dr. Prosper Mensah	CSIR-FORIG

Building Construction and Woodwork Technology	Isaac Buckman	Armed Forces Senior High Technical School
	Firmin Anewuoh	Presbyterian College of Education, Akropong-Akuapem
	Lavoe Daniel Kwaku	Sokode Senior High Technical School
Arabic	Dr. Mohammed Almu Mahaman	University for Development Studies
	Dr. Abas Umar Mohammed	University of Ghana
	Mahey Ibrahim Mohammed	Tijjaniya Senior High School
Art and Design Studio and Foundation	Dr. Ebenezer Acquah	University of Education Winneba
	Seyram Kojo Adipah	GES - Ga East Municipal Education Directorate
	Dr. Jectey Nyarko Mantey	Kwame Nkrumah University of Science and Technology
	Yaw Boateng Ampadu	Prempeh College
	Kwame Opoku Bonsu	Kwame Nkrumah University of Science and Technology
	Angela Owusu-Afriyie	Opoku Ware School
Aviation and Aerospace Engineering	Opoku Joel Mintah	Altair Unmanned Technologies
	David Kofi Oppong	Kwame Nkrumah University of Science and Technology
	Sam Ferdinand	Afua Kobi Ampem Girls' Senior High School
Biology	Paul Beeton Damoah	Prempeh College
	Jo Ann Naa Dei Neequaye	Nyakrom Senior High Technical School
	Abraham Kabu Otu	Prampram Senior High School
Biomedical Science	Dr. Dorothy Yakoba Agyapong	Kwame Nkrumah University of Science and Technology
	Davidson Addo	Bosomtwe Girls STEM SHS
	Jennifer Fafa Adzraku	
Business Management	Ansbert Baba Avole	Bolgatanga Senior High School
	Dr. Emmanuel Caesar Ayamba	Bolgatanga Technical University
	Faustina Graham	Ghana Education Service, HQ
Accounting	Nimako Osei Victoria	SDA Senior High School, Akyem Sekyere
	Emmanuel Kodwo Arthur	ICAG
	Bernard Adobaw	West African Examination Council
Chemistry	Awumbire Patrick Nsobila	Bolgatanga Senior High School

Chemistry	Paul Michael Cudjoe	Prempeh College
	Bismark Kwame Tunu	Opoku Ware School
	Michael Amissah	St. Augustine's College
Computing and Information Communication Technology (ICT)	Raphael Dordoe Senyo	Ziavi Senior High Technical School
	Kwasi Abankwa Anokye	Ghana Education Service, SEU
	Osei Amankwa Gyampo	Wesley Girls High School, Kumasī
	Dr. Ephriam Kwaa-Aidoo	University of Education Winneba
	Dr. Gaddafi Abdul-Salaam	Kwame Nkrumah University of Science and Technology
Design and Communication Technology	Gabriel Boafo	Kwabeng Anglican Senior High Technical School
	Joseph Asomani	Akenten Appiah Menka University of Skills Training and Entrepreneurial Development (AAMUSTED)
	Phyllis Mensah	Akenten Appiah Menka University of Skills Training and Entrepreneurial Development (AAMUSTED)
Economics	Dr. Peter Anti Partey	University of Cape Coast
	Charlotte Kpogli	Ho Technical University
	Salitsi Freeman Etornam	Anlo Senior High School
Engineering	Daniel Kwesi Agbogbo	Kwabeng Anglican Senior High Technical School
	Prof. Abdul-Rahman Ahmed	Kwame Nkrumah University of Science and Technology
	Valentina Osei-Himah	Atebubu College of Education
English Language	Esther Okaitsoe Armah	Mangoase Senior High School
	Kukua Andoh Robertson	Achimota School
	Beatrice Antwiwaa Boateng	Oti Boateng Senior High School
	Perfect Quarshie	Mawuko Girls Senior High School
French	Osmanu Ibrahim	Mount Mary College of Education
	Maurice Adjetey	Retired, CREF
	Mawufemor Kwame Agorgli	Akim Asafo Senior High School
General Science	Dr. Comfort Korkor Sam	University for Development Studies
	Robert Arhin	SDA Senior High School, Akyem Sekyere
Geography	Raymond Nsiah-Asare	Methodist Girls' High School

Geography	Prof. Ebenezer Owusu-Sekyere	University for Development Studies
	Samuel Sakyi-Addo	Achimota School
Ghanaian Languages	David Sarpei Nunoo	University of Education Winneba
	Catherine Ekua Mensah	University of Cape Coast
	Ebenezer Agyemang	Opoku Ware School
Government	Josephine Akosua Gbagbo	Ngleshie Amanfro Senior High School
	Augustine Arko Blay	University of Education Winneba
	Samuel Kofi Asafua Adu	Fettehman Senior High School
History	Dr. Anitha Oforiwah Adu-Boahen	University of Education Winneba
	Prince Essiaw	Enchi College of Education
Management in Living	Grace Annagmeng Mwini	Tumu College of Education
	Dorcas Akosua Opoku	Winneba Secondary School
Clothing and Textiles	Jusinta Kwakyewaa (Rev. Sr.)	St. Francis Senior High Technical School
	Rahimatu Yakubu	Potsin T.I Ahmadiyya SHS
Food and Nutrition	Ama Achiaa - Afriyie	St. Louis SHS
	Lily-Versta Nyarko	Mancell Girls' Senior High Technical School
Literature-in-English	Blessington Dzah	Ziavi Senior High Technical School
	Juliana Akomea	Mangoase Senior High School
Manufacturing Engineering	Benjamin Atribawuni Asaaga	Kwame Nkrumah University of Science and Technology
	Dr. Samuel Boahene	Kwame Nkrumah University of Science and Technology
	Ali Morrow Fatormah	Mfantsipim School
Mathematics	Edward Dadson Mills	University of Education Winneba
	Zakaria Abubakari Sadiq	Tamale College of Education
	Collins Kofi Annan	Mando Senior High School
Music	Pros Cosmas W. K. Mereku	University of Education Winneba
	Prof. Emmanuel Obed Acquah	University of Education Winneba
	Joshua Amuah	University of Ghana
	Benjamin Ofori	CRIG Primary School, Akim Tafo
	Davies Obiri Danso	New Juaben Senior High School
Performing Arts	Dr. Latipher Amma Osei Appiah-Agyei	University of Education Winneba
	Prof. Emmanuel Obed Acquah	University of Education Winneba
	Chris Ampomah Mensah	Bolgatanga Senior High School

Core Physical Education and Health	Dr. Mary Aku Ogum	University of Cape Coast
	Paul Kofi Yesu Dadzie	Accra Academy
Elective Physical Education and Health	Sekor Gaveh	Kwabeng Anglican Senior High Technical School
	Anthonia Afosah Kwaaso	Jukwa Senior High School
Physics	Dr. Linus Kweku Labik	Kwame Nkrumah University of Science and Technology
	Henry Benyah	Wesley Girls' High School, Cape Coast
	Sylvester Affram	Kwabeng Anglican Senior High School
Christian & Islamic Religious Studies	Dr. Richardson Addai-Mununkum	University of Education Winneba
	Dr. Francis Opoku	Valley View University College
	Dr. Francis Normanyo	Mount Mary College
	Dr. Haruna Zagoon-Sayeed	University of Ghana
	Kabiru Soumana	GES
	Seth Tweneboa	University of Education Winneba
Religious and Moral Education	Anthony Mensah	Abetifi College of Education
	Joseph Bless Darkwa	Volo Community Senior High School
	Clement Nsorwineh Atigah	Tamale Senior High School
Robotics	Dr. Eliel Keelson	Kwame Nkrumah University of Science and Technology
	Isaac Nzoley	Wesley Girls' High School, Cape Coast
Social Studies	Mohammed Adam	University of Education Winneba
	Simon Tengan	Wa Senior High Technical School
	Dr. Adwoa Dufie Adjei	University Practice Senior High School
	Dr. Isaac Atta Kwenin	University of Cape Coast
Spanish	Setor Donne Novieto	University of Ghana
	Franklina Kabio-Danlebo	University of Ghana
	Mishael Annoh Acheampong	University of Media, Art and Communication
Technical Support	Benjamin Sundeme	St. Ambrose College of Education
	Dr. Isaac Amoako	Atebubu College of Education
	Eric Abban	Mt. Mary College of Education

SECTION 1: BOOLEAN ALGEBRA

STRAND: COMPUTER ARCHITECTURE & ORGANISATION

Sub-Strand: Data Storage and Manipulation

Learning Outcome: *Develop knowledge and skills in implementing and applying programme execution and logic instructions effectively*

Content Standard: *Demonstrate knowledge and understanding of programme execution and logic instructions*

HINT

- ***Assign Portfolio Assessment** for the academic year by Week 2. Portfolio should be submitted by Week 22. **See Appendix A** of this Section and Teacher Assessment Manual and Toolkit pages 22-25 for more information on how to organise a portfolio assessment.*
- *Assign **Group Project** in Week 3. This should be submitted after Week 9. Refer to Teacher Assessment Manual and Toolkit pages 27-29 for how to conduct Project-Based Assessment. Refer to **Appendix B** which has been provided at the end of the section for the structure and rubrics of the group project.*

INTRODUCTION AND SECTION SUMMARY

This section introduces learners to computer architecture fundamentals, focusing on storing and manipulating data. They will learn basic ideas like storing data, organising memory, and the importance of logical rules for running programs. The content is meant to give learners the skills they need to get the most out of a programs performance and solve complex computational problems. By mastering these ideas, learners will be ready for further studies and career opportunities in computer science.

The section is interrelated with mathematics and electrical engineering. Teachers should intentionally emphasise the connections between these subjects to provide an understanding of how data storage and manipulation are integral to the broader field of computer science.

The weeks covered by the sections are:

Week 1: Determine critical points of logic operations and apply them to solve problems efficiently.

Week 2: Apply the concepts of Boolean expressions to simplify circuits.

Week 3: Discuss 3 major arithmetic operations and apply them, considering their various functions.

SUMMARY OF PEDAGOGICAL EXEMPLARS

This section employs diverse teaching strategies to ensure a comprehensive understanding, critical thinking, and peer learning. Some lessons begin with direct instruction to explain key concepts. This section adopts collaborative learning in mixed-ability groups promoting peer teaching to enhance understanding. Reflective discussions on fundamental logic operations using truth tables and circuit diagrams are conducted to deepen comprehension. Through experiential learning, learners identify critical points within logical operations. Q&A sessions led by the teacher are used to evaluate learners' understanding.

ASSESSMENT SUMMARY

This section uses depth of knowledge (DOK) to improve the quality, validity, and fairness of assessments while also encouraging learners to learn more deeply and think critically. The DOK levels ensure that tests correctly measure how well learners understand the content and give teachers the information they can use to plan and help learners understand the focal areas better.

To ensure that all of a learner's abilities are evaluated so that skills and areas that need improvement can be found and used to plan future lessons, teachers are to use the weekly tasks and assessment questions to give both formative and summative tests on the topics covered in "Boolean Algebra."

It is important to note that the test items in this manual only serve as a guide for the teacher to establish learners' understanding of the lessons taught. They do not limit the teacher from exploring and creating innovative questions.

WEEK 1

Learning Indicator: Determine critical points of logic operations and apply them to solve problems efficiently

FOCAL AREA: UNDERSTANDING LOGIC OPERATIONS AND EXPRESSIONS

1. Recap AND, OR and NOT operations to ensure learners are comfortable with these operations and their truth tables.
2. **NAND Gate (Not AND)**

   The NAND gate is a primitive logic gate made by the combination of the AND gate and the NOT gate, connected in series. It behaves in an opposite manner to the AND gate. The truth table for the NAND gate is shown in Figure 1.1.

A	B	A AND B	Apply Not(~)	NAND
0	0	0	~ 0	1
0	1	0	~ 0	1
1	0	0	~ 0	1
1	1	1	~ 1	0

Figure 1.1: *NAND Truth Table*

3. **NOR Gate (Not OR)**

   The NOR gate is a primitive logic gate made by the combination of the OR gate and the NOT gate. The truth table for the NOR gate is shown in Figure 1.2.

A	B	A or B	Apply Not(~)	NOR
0	0	0	~ 0	1
0	1	1	~ 1	0
1	0	1	~ 1	0
1	1	1	~ 1	0

Figure 1.2: *NOR Truth Table*

4. **XOR Gate (eXclusive OR)**

   The XOR gate combines (A AND NOT B) OR (NOT A AND B). Figure 1.3 shows the truth table of the XOR gate.

A	~A	B	~B	~A and B	A and ~B	XOR
0	1	0	1	0	0	0
0	1	1	0	1	0	1
1	0	0	1	0	1	1
1	0	1	0	0	0	0

Figure 1.3: *XOR truth table*

TO DO

Use of NAND and NOR Gates to Create Other Logic Gates

One interesting aspect of logic gates is that you can construct any other logic gate using just NAND or NOR gates.

Examples creating logic gates with NAND gates

NOT Gate: A NOT gate can be created by connecting both inputs of a NAND gate to the same input signal.

AND Gate: To create an AND gate, first create a NAND gate and then pass its output through a NOT gate.

OR Gate: To create an OR gate using NAND gates, first invert both inputs using NAND gates as NOT gates, then pass the results through another NAND gate.

Fun Challenge

Challenge learners to create the NOT, AND, and OR logic gates using NOR gate.

Challenge learners to create the other logic gates; XOR and XNOR using only NAND or NOR gates.

This exercise helps learners understand the versatility and importance of these gates in digital circuits.

5. **Standard symbols of Logic gates**

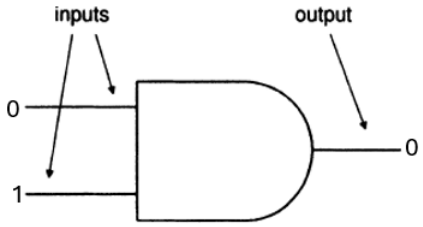


Figure 1.4: *Input and output labels of a logic gate*

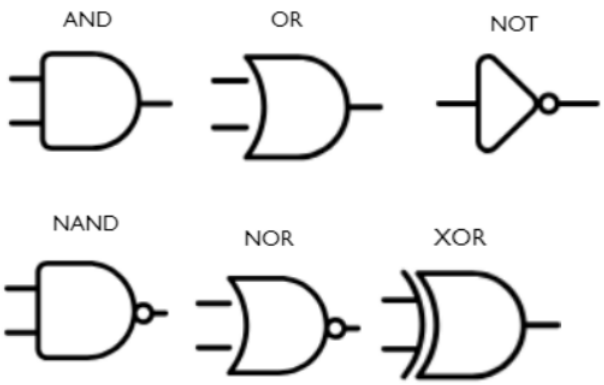


Figure 1.5: *Symbols of logic gates*

6. **Combinational Gates**

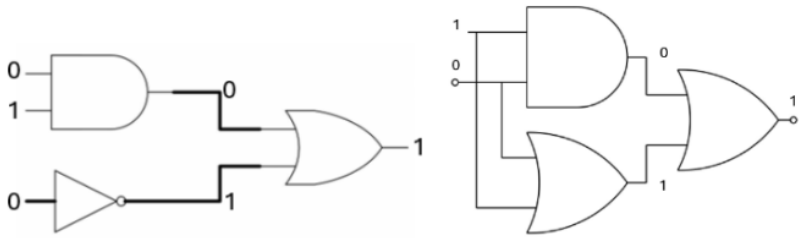


Figure 1.6: *Circuits*

The first circuit combines an AND gate with inputs 0 and 1, producing 0, which is then used with a 1 from the NOT gate in an OR gate, resulting in a final output of 1.

The second circuit starts with an AND gate with inputs 1 and 0, resulting in 0; again, the inputs from the AND gates are used in the first OR gate, producing 1. The outs 0 and 1 are input in another OR gate, resulting in 1 as the final result.

7. **Identifying intermediate stages**

The intermediate stages for the gates are the partial results calculated when adding the inputs. We use them to understand the flow of logic within the gate, enabling efficient design, analysis and troubleshooting.

For example,

- The intermediate stages in Figures 1.1 and 1. 3 are "A AND B" and "(~A AND B), (A AND~B)", respectively.
- In the first circuit in Figure 1.6, the AND and OR gates are intermediate to the final result.

LEARNING TASKS

1. Learners discuss and complete truth tables for each logic gate including AND, OR and NOT.
2. Learners create truth tables for NAND, NOR and XOR gates.
3. Learners observe symbols of various logic gates facilitated by the teacher.
4. Learners match symbols to their corresponding gates (AND, OR, NOT, NAND, NOR, XOR).
5. Learners solve NAND, NOR, and XOR logic questions using truth tables.
6. Learners connect logic gates in a variety of circuits and explain their solutions.
7. Learners annotate the inputs, outputs, and intermediate stages.
8. Learners design a logic circuit to perform specific tasks and explain their solutions.

PEDAGOGICAL EXEMPLARS

1. **Reflective learning:** The teacher will lead the class in an open discussion using truth tables to show the logic operations of AND, OR, and NOT.
2. **Experiential Learning**
   a. The teacher guides learners using truth tables to explain NAND, NOR and XOR logic.
   b. The teacher guides learners in identifying the symbols used in circuits.
   c. The teacher will guide learners in identifying critical points within logical operations where decisions or outcomes are determined (highlighting the significance of inputs, outputs, and intermediate stages in logical processes.)
3. **Collaborative Learning**
   a. Teachers supervise learners in their groups to solve questions about NAND, NOR and XOR logic using truth tables.
   b. Teachers supervise learners in their mixed-ability groups to connect logic gates in a circuit.
   c. Teachers supervise learners in mixed-ability groups to show the inputs, outputs, and intermediate stages.
4. **Problem-Based Learning and Group Presentation**
   a. The teacher will present a scenario where learners must design a logic circuit to perform a specific task.
   b. The teacher gives learners equal chances to explain their solutions.
   c. The Teacher intervenes and facilitates areas that need clearer explanations.

KEY ASSESSMENT

DoK Level 1: Recall

1. Identify the following symbols with their gate names

DoK Level 2: Skills/Concept Application

Construct a logic circuit using only NAND gates that acts as an AND gate with two inputs (A and B). Provide the truth table for your circuit and explain how the combination of NAND gates produces the same output as an AND gate.

Instructions

- Draw the circuit diagram using only NAND gates
- Label all inputs and outputs clearly
- Create a truth table with all possible combinations of inputs (A and B)
- Explain the logic behind your circuit design

DoK Level 3: Strategic Thinking

1. You are designing a programme to identify even numbers. How can you efficiently use logic operations to determine even numbers without unnecessary calculations?
2. Design a logic circuit that takes two inputs (A and B) and produces an output that is true (1) only when the inputs are different (i.e., one is true and the other is false). You may use any combination of AND, OR, NOT, and XOR gates in your design. Provide the truth table for your circuit, including a detailed explanation of your design process, and justify your choice of logic gates.

DoK Level 4: Extended Thinking

1. A complex programme checks for multiple conditions before acting. How can you analyse the logic flow and identify opportunities to optimise the programme by rearranging or combining logic operations?
2. Design a programme that uses efficient logic operations to solve a traffic light control problem.
3. Design a programme to identify even numbers with the use of logic operations.

HINT

The recommended mode of assessment for the week is demonstration, using DoK level 4 question 3 as a sample. Refer to the Teacher Assessment Manual and Toolkit for additional information about how to administer this assessment mode.

WEEK 2

Learning Indicator: Apply the concepts of Boolean expressions to simplify circuits

FOCAL AREA: APPLYING BOOLEAN EXPRESSIONS

1. **Notation:** Boolean operators AND, OR and NOT have symbolical representations as ∧, ∨ and ¬ respectively.
2. The term **Normal form** in logic refers to a standardized way of representing Boolean functions or logical expressions.
3. **Identity Law:** This law states that a variable would remain unchanged when it is ANDed with '1' or ORed with '0',

   i.e.,

   A.1 = A

   A + 0 = A
4. **De Morgan Law**: This law states that the operation of an OR or an AND logic circuit stays unchanged whenever all inputs are inverted, while the operator changes from AND to OR whenever the outputs are inverted, i.e.,

   (A.B)' = A' + B'

   (A + B)' = A'.B'
5. **Disjunctive Normal Form (DNF):** DNF is a standard way of expressing a Boolean function. It is a disjunction (OR) of one or more conjunctions (ANDs) of variables.

   Example

   a. A ∧ (B ∨ C) = (A ∧ B) ∨ (A ∧ C)

      Note that the emphasis is on distributing the A over the OR operation to have the OR(s) between the bracket(s) and the AND(s) inside the brackets.
6. **Conjunctive Normal Form (CNF):** CNF is another standard way of expressing a Boolean function. It is a conjunction (AND) of one or more disjunctions (ORs) of literals.

   Example

   a. A ∧ (B ∨ C) = (A ∨ B) ∧ (A ∨ C)

   Note that the emphasis is on distributing the A over the AND operation.
7. **Simplifying Circuits with Boolean Expressions**

   Simplifying digital circuits with DNF or CNF decreases digital circuit complexity. This produces ideal designs that are simpler to use and troubleshoot.

   Example: Look at how complex the circuit in Figure 2.1 is and how simple Figure 2.2 is after using DNF and CNF to simplify the circuit. For the same expression: Y = (A OR B) AND (A OR -B) AND (A OR C)

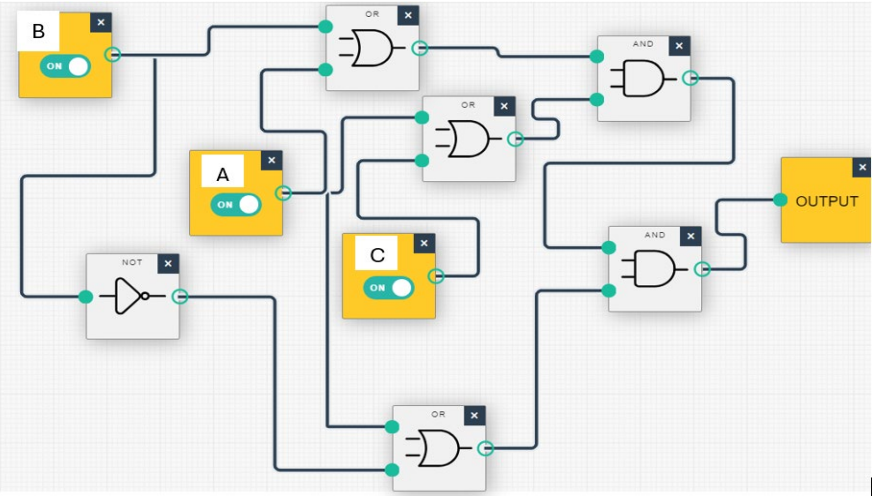


Figure 2.1: *Complex Circuit*

Simplification process using DNF and CNF

Y = (A OR B) AND (A OR -B) AND (A OR C)

Y = A OR (B AND -B AND C)

Y = A OR (0 AND C) Where: B AND -B = 0

Y = A OR (0) Where: 0 AND C = 0

Y = A Where: A OR 0 = A

Therefore, Y = A

Figure 2.2: *Simplified circuit*

LEARNING TASKS

1. Learners link the signs (∧, ∨, ¬) to (AND, OR, NOT).
2. Learners solve question(s) involving the use of the symbols.
3. Learners link their understanding of disjunction and conjunction in maths to solve some distributive questions.
4. If (c) is not applicable, learners should observe how the computing teacher facilitates the DNF and CNF (distributive operation).
5. Learners solve distributive questions.
6. Learners draw a circuit for a given expression (like the example above).
7. Learners simplify a given circuit and draw another circuit for their simplified expression.

8. Learners use online logic simulators to draw circuit.

PEDAGOGICAL EXEMPLARS

1. **Experiential Learning**
   a. The teacher explains the logical symbols (∧, ∨, ¬) and their corresponding operations (AND, OR, NOT). Individual Practice.
   b. The teacher guides learners in solving questions, individually, that involve using the symbols ∧, ∨, ¬.
2. **Direct Instruction**
   a. Resource Personnel (if applicable): Under the guidance of a maths teacher, learners link their understanding of disjunction (OR) and conjunction (AND) in mathematics to solve distributive questions.
   b. The computing teacher demonstrates converting Boolean expressions into DNF and CNF forms.
3. **Collaborative Learning**
   a. The teacher guides learners to work in mixed-ability pairs to solve distributive questions under the teacher's guidance.
   b. Teacher combines paired learners to form larger groups, pairing proficient learners with those who need more help, discussing the answers for a.
   c. The teacher facilitates group tasks in mixed-ability groups, where learners draw a circuit for a Boolean expression.
   d. The teacher facilitates a group task in which learners simplify the given Boolean expression and redraw the circuit for the simplified expression.
   e. Teachers can guide learners in using online logic simulators to draw logic gates (if applicable).
4. **Explorative Learning**: The teacher guides learners in comparing their original and simplified circuit drawings from the collaborative learning activities and writing down their findings on using DNF and CNF in simplifying circuits.

KEY ASSESSMENT

The summative assessment questions below are only to serve as a guide for the teacher when creating questions to measure learners' comprehension of the focal area.

DoK Level 1: Recall

Identify the correct symbolic representation for the Boolean operators AND, OR, and NOT from the following options:

a) ∧ for AND, ∨ for OR, ¬ for NOT
b) ∨ for AND, ∧ for OR, ¬ for NOT
c) ∧ for OR, ∨ for AND, ¬ for NOT
d) ∨ for NOT, ∧ for AND, ¬ for OR

DoK Level 2: Skills/Concept Application

Convert the following Boolean expression into Disjunctive Normal Form (DNF):

$$(A \wedge B) \vee C$$

Provide the step-by-step process you used to convert the expression.

DoK Level 3: Strategic Thinking

Given the Boolean expression (A ∨ B) ∧ (A ∨ ¬B) ∧ (A ∨ C), simplify the expression using the Disjunctive Normal Form (DNF). 5 marks

DoK Level 4: Extended Thinking

Analyze the impact of using Disjunctive Normal Form (DNF) and Conjunctive Normal Form (CNF) on simplifying complex digital circuits. Choose a complex Boolean expression, convert it to both DNF and CNF.

HINT

Assign Portfolio Assessment *for the academic year by the end of Week 2. Portfolio should be submitted by Week 22.* ***See Appendix A*** *of this Section and Teacher Assessment Manual and Toolkit pages 22–25 for more information on how to organise a portfolio assessment.*

WEEK 3

Learning Indicator: Discuss 3 major Arithmetic Operations and apply them, considering their various functions

FOCAL AREA: ARITHMETIC OPERATIONS

1. **Explanation of the Arithmetic Operations**
   a. **Addition (+):** Addition combines two or more numbers to get their total.

      Example

      i. $5 + 3 = 8$
      ii. $5 + 3 = 8$

      Properties to note

      i. addition is commutative, meaning: a + b is the same as b + a.
      ii. addition is associative, meaning: a + (b + c) = (a + b) + c.
      iii. Identity: 0 is the identity for addition $a + 0 = a$.

   b. **Subtraction (-):** Subtraction is finding the difference between two numbers.

      Example

      i. i. $5 - 3 = 2$
      ii. 3 -5 = -2

      Properties

      i. **Non- Commutative :** $a - b \neq b - a$
      ii. **Associative :** $(a - b) - c \neq a - (b - c)$

   c. **Multiplication (x)** means combining several groups of the same size. It is a faster and more efficient way to find the sum than adding the same number repeatedly.

      In computing, "*" is the symbol used for multiplication but not "x".

      Example

      i. 5∗3=15
      ii. 3∗5=15

      Properties

      i. Commutative: a∗b = b∗a
      ii. Associative: (a∗b)∗c = a∗(b∗c)
      iii. Identity: a∗1 = 1∗a = a
      iv. Distributive: (b + c) = (a * b) + (a * c)

   d. **Division (/ or ÷):** Means splitting a number into parts.

      Example

      i. 6/3=2

Properties

i. Non-Commutative: a/b ≠ b/a

ii. Associative: (a/b)/ c ≠ a/(b/c)

iii. Identity: a/1=a but not 1/a

iv. Distributive: a / (b * c) = (a / b) * (a / c)

e. **PEDMAS:** PEMDAS is an acronym that helps you remember the order of operations in arithmetic and algebra. Thanks to the order of operations, mathematical expressions are evaluated consistently and accurately. PEMDAS means P—parentheses, E—exponents, M—multiplication, D—division, A—addition, and S—subtraction.

Application

To ensure accuracy, evaluate expressions according to the PEMDAS-outlined order of operations.

Example: 3+6×(5+4)÷3^2−1. The final result is 8.

2. **Implementation of arithmetic operations in Python**

Teachers can use any Python compiler, such as to facilitate the lesson.

Scan this QR code to get access to some codes to facilitate learning.

LEARNING TASKS

1. Learners take notes and ask questions to clarify their understanding.
2. Learners create a chart listing each operation with its properties and examples.
3. Learners observe the coding process as the teacher demonstrates.
4. Learners write code snippets on numeric data types for addition, subtraction, multiplication, and division—example code snippets in Python.
5. Learners, in their mixed-ability groups, write simple steps to calculate totals, averages, and percentages for a set of data given by the teacher.
6. Learners ask open-ended questions about arithmetic operations, their functions, and use cases.

PEDAGOGICAL EXEMPLARS

1. **Direct Instruction**
   a. The teacher explains the concepts of addition, subtraction, multiplication, and division. Use examples to illustrate each operation, emphasising the symbols.
   b. The teacher demonstrates how arithmetic operations are implemented in a programming language (e.g., Python)
   c. The teacher should discuss use cases of arithmetic operations in computing, such as mathematical calculations, data manipulation, and algorithmic tasks.
2. **Collaborative Learning:** The teacher provides learners with the task of writing their code snippets for arithmetic operations. This should be done in mixed-ability groups.
3. **Problem-Based Learning**: The teacher will present a scenario where learners must develop a programme to calculate totals, averages, and percentages for a data set.
4. **Open Class Discussion**: The teacher will lead a discussion with the learners so they can ask open-ended questions about arithmetic operations, their functions, and use cases.

KEY ASSESSMENT

DoK Level 1: Recall

1. What are the three main arithmetic operations?
2. Briefly describe what each of these operations do

DoK Level 2: Skills/Concept Application

Identify the appropriate operation (addition, subtraction, multiplication) to find the total cost of items.

DoK Level 3: Strategic thinking

1. You are calculating the total cost of a project, including materials and labour. How can you combine addition and multiplication to find the final cost?
2. Describe a real-world situation where the order of operations (PEMDAS) is important to get the correct answer (e.g., calculating average speed with distance and time)
3. Formative Assessment: Learning Task 3a.
4. Case

   The Opoku family is planning a day out to the Accra Zoo. They need to calculate the total cost of the trip to ensure it fits within their budget. The cost includes entrance fees, food, transportation and souvenirs. The family decides to calculate the cost using the following information:

   i. Entrance fee per person: GHS 50
   ii. Food per person: GHS 30
   iii. Transportation cost (fixed for the whole family): GHS 100
   iv. Souvenirs (fixed for the whole family): GHS 60

   The family consists of 2 adults and 3 children. Mr. Opoku decides to calculate the total cost using the following expression:

   Total Cost= (Entrance fee per person + Food per person) × Number of people + Transportation cost + Souvenirs

a. Use the order of operations (PEMDAS) to correctly calculate the total cost for the Opoku family's day out.

b. Explain why the order of operations is important in this case to ensure the correct calculation.

DoK Level 4: Extended Thinking

Explain the approach and the mathematical operations involved in solving unit conversions efficiently.

HINT

It is recommended that teachers give learners a Group Project work on "designing a plan for implementing a barcode system in a new business". This is to be submitted in week 9. Refer to Appendix B for more information on the group project work.

SECTION 1 REVIEW

In this section, learners explore the fundamentals of computer architecture, focusing on data storage, memory organisation, and logical rules for programme execution. They acquire skills to optimise programme performance and tackle complex computational problems, preparing them for advanced studies and careers in computer science. The content connects with mathematics and electrical engineering. Weekly topics include logic operations (Week 1), Boolean expressions for circuit simplification (Week 2), and arithmetic operations (Week 3). Diverse teaching strategies and Depth of Knowledge (DOK) assessments ensure comprehensive understanding and critical thinking.

APPENDIX A: PORTFOLIO

1. ***Task:*** Collect all your work from the academic year, starting from the beginning of the first semester, and compile it into a portfolio to be submitted at the end of the year for assessment. Your portfolio should include assignments, projects, quizzes, tests, reflective pieces, and mid-term and end-of-semester papers. This portfolio will be assessed to evaluate your overall understanding and progress throughout the year.
2. ***Artefacts:*** Example of learners' works to be included in the Portfolio Assessment
    a. Assignments
    b. Projects
    c. Quizzes and Tests
    d. Reflective Pieces
    e. Mid semester and end of semester papers
3. ***Organisation/Structure***
    a. Cover Page which entails the learner's name, class, subject and period/date.
    b. Table of Contents which has a list of items included with page numbers.
    c. Brief description/background of items such as short description of the significance of sports certificates and awards, background information for each included artefact etc.
4. ***Mode of Administration***
    a. Explain the purpose and components of the portfolio to the learners and provide examples and templates for each section.
    b. Schedule periodic reviews (e.g., every 3-4 weeks) to ensure learners are keeping up with their portfolios and provide feedback and guidance during these checkpoints.
    c. Provide learners with the scoring rubrics and provide detailed explanation on the rubrics.
    d. Final portfolios are due in week 22 of the academic calendar. Allow a grace period for revisions based on final feedback.
5. ***Sample mode of submission/presentation***
    a. Communicate the final deadline for portfolio submission to all learners to ensure timely and complete submissions.
    b. Learners will submit their completed portfolios either as a physical or through the school's online submission system.
    c. Ensure the portfolio includes all required elements: assignments, projects, quizzes, tests, reflective pieces, class participation records, and a final reflection.
    d. Learners should organise their portfolios clearly and logically, with each component clearly labelled and easy to access.
    e. For digital submissions, learners should upload their portfolios as a single file or in clearly marked folders within the online portal.

6. *Feedback Strategy*

a. Schedule periodic check-ins to discuss progress, set goals, and adjust strategies as needed.

b. Utilise both formative and summative feedback to guide learners' development and ensure they understand how to enhance their work continuously.

7. *Sample Portfolio Assessment Marking scheme*

Learner's works	*Score*
Assignments/Exercises	*10 marks*
Projects/Case studies	*10 marks*
Quizzes and Tests	*10 marks*
Reflective Pieces	*5 marks*
Mid-semester and End-of-semester Papers	*5 marks*
Total marks	*40 marks*

APPENDIX B: GROUP PROJECT WORK

Assessment Task

In groups, design a comprehensive plan for implementing a barcode system in a new business. The plan should include a detailed explanation of the barcode system, the necessary resources, training for staff, and a timeline for implementation

Rubrics for Group Project Work

Criteria	Excellent (4)	Good (3)	Satisfactory (2)	Needs Improvement (1)
Critical Thinking	*Showing 4 of the skills:* *1. Understanding of Barcode System:* *Explanation of barcode technology (e.g., linear barcodes, QR codes),* *uses in inventory management and customer tracking.* *2. Analysis of Necessary Resources:* *Barcode scanners, printers, software, and databases.* *3. Staff Training Plan:* *Content of the training (e.g., how to use scanners, troubleshooting, and software navigation)* *4. Implementation Timeline:* *procurement, setup, testing, rollout.* *specific timeframes for each phase.* *Problem-Solving Approach:* *Challenges like technical issues, staff resistance, or budget overruns.* *Solutions like contingency funds, IT support, or ongoing staff engagement.*	*Showing 3 of the skills:* *1. Understanding of Barcode System:* *Explanation of barcode technology (e.g., linear barcodes, QR codes), uses in inventory management and customer tracking.* *2. Analysis of Necessary Resources:* *Barcode scanners, printers, software, and databases.* *3. Staff Training Plan: Content of the training (e.g., how to use scanners, troubleshooting, and software navigation).* *4. Implementation Timeline:* *procurement, setup, testing, rollout, specific timeframes for each phase.* *5. Problem-Solving Approach:* *Challenges like technical issues, staff resistance, or budget overruns.* *Solutions like contingency funds, IT support, or ongoing staff engagement.*	*Showing 2 of the skills:* *1. Understanding of Barcode System:* *Explanation of barcode technology (e.g., linear barcodes, QR codes), uses in inventory management and customer tracking.* *2. Analysis of Necessary Resources:* *Barcode scanners, printers, software, and databases.* *3. Staff Training Plan: Content of the training (e.g., how to use scanners, troubleshooting, and software navigation).* *4. Implementation Timeline:* *procurement, setup, testing, rollout, specific timeframes for each phase.* *5. Problem-Solving Approach:* *Challenges like technical issues, staff resistance, or budget overruns.* *Solutions like contingency funds, IT support, or ongoing staff engagement.*	*Showing 1 of the skills:* *1. Understanding of Barcode System:* *Explanation of barcode technology (e.g., linear barcodes, QR codes), uses in inventory management and customer tracking.* *2. Analysis of Necessary Resources:* *Barcode scanners, printers, software, and databases.* *3. Staff Training Plan:* *Content of the training (e.g., how to use scanners, troubleshooting, and software navigation).* *4. Implementation Timeline:* *procurement, setup, testing, rollout.* *Specific timeframes for each phase.* *5. Problem-Solving Approach:* *Challenges like technical issues, staff resistance, or budget overruns.* *Solutions like contingency funds, IT support, or ongoing staff engagement.*

Collaboration	Exhibit 4 of these Contributing to the group. Respecting the views of others Tolerating others Resolving conflicts Taking responsibility	Exhibit 3 of these Contributing to the group. Respecting the views of others Tolerating others Resolving conflicts Taking responsibility	Exhibit 2 of these Contributing to the group. Respecting the views of others Tolerating others Resolving conflicts Taking responsibility	Exhibit 1 of these Contributing to the group. Respecting the views of others Tolerating others Resolving conflicts Taking responsibility
Communication	Showing 4 of the skills e.g. 1. Audible voice 2. Keeping eye contact 3. Pay attention to audience 4. Engaging the audience with interaction 5. Use of gesture	Showing 4 of the skills e.g. 1. Audible voice 2. Keeping eye contact 3. Pay attention to audience 4. Engaging the audience with interaction 5. Use of gesture	Showing 4 of the skills e.g. 1. Audible voice 2. Keeping eye contact 3. Pay attention to audience 4. Engaging the audience with interaction 5. Use of gesture	Showing 4 of the skills e.g. 1. Audible voice 2. Keeping eye contact 3. Pay attention to audience 4. Engaging the audience with interaction 5. Use of gesture
Problem Solving	Showing 4 of the skills: E.g. 1. Understanding the Problem: Inventory management, efficiency, error reduction, real-time tracking, customer service improvement. 2. Proposing Solutions: Barcode scanners, point-of-sale systems, cloud-based storage, integration with inventory software, workflow automation. 3. Evaluating Resources: Hardware (barcode readers, printers), software (inventory management systems), budget, technical support, internet connectivity.	Showing 4 of the skills: E.g. 1. Understanding the Problem: Inventory management, efficiency, error reduction, real-time tracking, customer service improvement. 2. Proposing Solutions: Barcode scanners, point-of-sale systems, cloud-based storage, integration with inventory software, workflow automation. 3. Evaluating Resources: Hardware (barcode readers, printers), software (inventory management systems), budget, technical support, internet connectivity.	Showing 4 of the skills: E.g. 1. Understanding the Problem: Inventory management, efficiency, error reduction, real-time tracking, customer service improvement. 2. Proposing Solutions: Barcode scanners, point-of-sale systems, cloud-based storage, integration with inventory software, workflow automation. 3. Evaluating Resources: Hardware (barcode readers, printers), software (inventory management systems), budget, technical support, internet connectivity.	Showing 4 of the skills: E.g. 1. Understanding the Problem: Inventory management, efficiency, error reduction, real-time tracking, customer service improvement. 2. Proposing Solutions: Barcode scanners, point-of-sale systems, cloud-based storage, integration with inventory software, workflow automation. 3. Evaluating Resources: Hardware (barcode readers, printers), software (inventory management systems), budget, technical support, internet connectivity.

	4. Considering Staff Training: Hands-on training, system manuals, troubleshooting skills, role-specific sessions, continuous learning. *5. Creating a Timeline:* *Planning phase, resource acquisition, system installation, training, testing and rollout, post-implementation review*	*4. Considering Staff Training: Hands-on training, system manuals, troubleshooting skills, role-specific sessions, continuous learning.* *5. Creating a Timeline:* *Planning phase, resource acquisition, system installation, training, testing and rollout, post-implementation review.*	*4. Considering Staff Training: Hands-on training, system manuals, troubleshooting skills, role-specific sessions, continuous learning.* *5. Creating a Timeline:* *Planning phase, resource acquisition, system installation, training, testing and rollout, post-implementation review*	*4. Considering Staff Training: Hands-on training, system manuals, troubleshooting skills, role-specific sessions, continuous learning.* *5. Creating a Timeline:* *Planning phase, resource acquisition, system installation, training, testing and rollout, post-implementation review*

Total = 16 mark

SECTION 2: FUNCTIONALITIES OF HARDWARE, SOFTWARE AND BARCODES

STRAND: COMPUTER ARCHITECTURE & ORGANISATION

Sub-Strand: Computer Hardware and Software

Learning Outcome: *Develop knowledge and skills in implementing and applying programme execution and logic instructions effectively*

Content Standard: *Demonstrate knowledge and understanding of programme execution and logic instructions*

HINT

Remind learners of Mid semester examination in week 6. Refer to the Appendix C for a Table of Specification to guide you to construct your test items.

INTRODUCTION AND SECTION SUMMARY

In this section, learners will explore the essential aspects of computer architecture and organisation, focusing on the functionality of hardware, software, and barcodes. By examining computer hardware components and various types of software, learners will develop a comprehensive understanding of input and output devices. Emphasis will be placed on creating and utilising barcodes and QR codes using scanning tools. Additionally, learners will apply their acquired skills and knowledge to explain the operation of these functionalities and the overall functionality of an operating system to enhance their ability to control and monitor digital systems effectively.

The weeks covered by the section are:

Week 4: Demonstrate creating and utilising barcodes and QR codes using input and output devices, such as scanners, digital cameras, keyboards, microphones, mice, 2D/3D scanners, and touch-screen sensors for control and monitoring

Week 5: Demonstrate creating and utilising barcodes and QR codes using input and output devices, such as scanners, digital cameras, keyboards, microphones, mice, 2D/3D scanners, and touch-screen sensors for control and monitoring

Week 6: Explain the types of software and their functions

PEDAGOGICAL EXEMPLAR SUMMARY

In this section, the teacher uses various methods to teach barcode hardware and software functionalities. Most weeks begin with a presentation and discussion. Learners explore the importance of focal areas through guided discussions and targeted questions. In a research-based learning activity, learners in mixed-ability groups investigate various functionalities. Finally, in project-based learning, learners create presentations to share their findings with the wider class to enhance their understanding through practical application and peer collaboration.

ASSESSMENT SUMMARY

This section uses Depth of Knowledge (DOK) to encourage learners to think critically and learn more deeply while enhancing assessments' quality, validity, and fairness.

Teachers are expected to use the weekly tasks and assessment questions to give both formative and summative tests on the material covered in "Functionalities of Hardware, Software, and Barcodes" to ensure that every learner's abilities are assessed. This will allow skills and areas that require improvement to be identified and used to plan future lessons.

It is important to note that the test items in this manual only serve as a guide for the teacher to establish learners' understanding of the lessons taught. They do not limit the teacher from exploring and creating innovative questions.

WEEK 4

Learning Indicator: Demonstrate creating and utilising barcodes and QR codes using input and output devices, such as scanners, digital cameras, keyboards, microphones, mice, 2D/3D scanners, and touch-screen sensors for control and monitoring

Focal Area: CREATING AND UTILISING BARCODES AND QR CODES

Barcodes and QR codes efficiently and rapidly store and retrieve information. They are used for information exchange, tracking, and inventory management in various industries. Utilising input and output devices like scanners, digital cameras, and touch-screen sensors enhances their application.

1. **Barcode**

   A barcode is a symbol that a smartphone app or barcode scanner can easily read. Then, a link between a product and its database information must be established.

   It is important to know that each barcode is a globally unique number. They are commonly used in retail stores at the checkpoint. Retail stores use barcodes to quickly retrieve a product's name and price at checkout and track products as they move through the supply chain. Once scanned, store owners will know which items were purchased more frequently, which were not, or which were out of stock.

   Types of barcodes

   a. **EAN-13 Barcodes / International Article Number**: International Article Numbers, formerly European Article Numbers (EAN), consist of 13 digits. Every EAN-13 barcode is encoded into a barcode image and is unique to a particular product. When the barcode is scanned into a store's system, the barcode records price and inventory data related to the product. This means that every product has a unique EAN-13 number.

      These barcodes are used globally on all retail products except books and magazines and in the USA and Canada, where UPC-A (Universal Product Codes) barcodes are more prevalent.

Figure 4.1: *IAN barcode*

   b. **Universal Product Codes (UPC-A Barcodes)**: These 12-digit barcodes are commonly found on retail goods in the USA and Canada, except for books and magazines.

Figure 4.2: *UPC-A Barcode*

c. **ITF-14 Carton & Pallet Barcodes**: ITF-14 Carton Codes are created from EAN-13 and UPC-A barcodes. They are not used for individual products (UPC-A or EAN) but only on cartons containing specific quantities of the products.

Figure 4.3: *ITF-14 Carton*

d. **ISBN Book Barcodes:** International Standard Book Numbers (ISBN) are barcodes for hard copies and e-books. The ISBN agencies in each country are responsible for distributing ISBNs. They use standard retail barcodes (EAN or UPC) beginning with the number 978.

   Note that crossword, puzzle, and colouring books do not use ISBN barcodes because they are not published.

Figure 4.5: *ISBN Barcode*

e. **ISSN Magazine Barcodes:** An 8-digit International Standard Serial Number (ISSN) uniquely identifies publications such as magazines and journals.

Figure 4.6: *ISSN Barcode*

2. **Creating a barcode**

Creating a barcode for your product follows the following steps:

a. Determine whether you need EAN-13 or UPC barcodes based on where your product will be sold.
b. Consider product variations and determine the number of barcodes you require. Keep in mind that different flavours, sizes, and colours require different barcodes.
c. After step b, buy your item's UPC or EAN barcode online in the appropriate quantity.
d. You will receive an email with the barcode number, numbers, and files in the desired file format that includes the corresponding images.
e. Add the purchased barcode to your product (take into consideration resizing to fit your product packaging and ensure the barcode is in an easily visible, flat location).

LEARNING TASKS

1. Learners watch a slide presentation on how barcodes are used in retail stores, warehouses, books and magazines.
2. Learners discuss the importance of the use of barcode in the presentation.
3. Learners research the types of barcodes and categorise them into retail book and magazine barcodes. (Scan for reading resources).

4. Learners make presentations on how to create barcodes.

PEDAGOGICAL EXEMPLARS

1. **Presentation and Discussion**

a. The teacher presents slides on how barcodes are used in retail stores, warehouses, and on books and magazines.
b. The teacher facilitates discussion on the different uses of barcodes presented in the video.
c. The teacher guides learners to tease out the importance of barcodes in retail and warehouse management.
d. The teacher questions why barcodes help inventory control for books and magazines.

2. **Research-Based Learning:** The teacher tasks learners in their mixed-ability groups to research various types of barcodes and categorise them into retail book and magazine barcodes.
3. **Project-Based Learning:** The teacher tasks learners, in their groups, to create presentations on how to create barcodes. Each group is assigned a different barcode type.

KEY ASSESSMENT

DoK Level 1: Recall

What are the primary uses of barcodes, as shown in the video presentation?

DoK Level 2: Skills/Concepts Application

Explain how barcodes improve efficiency in retail stores and warehouses.

DoK Level 3: Strategic thinking

1. Discuss the impact of barcode technology on inventory management and customer service.
2. Conduct research on how barcode technology has influenced inventory management and customer service. Consider how businesses use barcodes to track products, manage stock levels, and enhance the customer shopping experience. Write a brief report summarising your findings.

Level 4: Expanded Thinking

Design a comprehensive plan for implementing a barcode system in a new business, including training and resource allocation.

HINT

This week's recommended mode of assessment is research using Dok level 2 question as a sample. Refer to the Teacher Assessment Manual and Toolkit page 151 for information on how to use this mode of assessment.

WEEK 5

Learning Indicator: Demonstrate creating and utilising barcodes and QR codes using input and output devices, such as scanners, digital cameras, keyboards, microphones, mice, 2D/3D scanners, and touch-screen sensors for control and monitoring

FOCAL AREA: BARCODES AND QR CODES

1. **QR Codes (Quick Response Codes)** are barcodes that can store a significant amount of data compared to traditional barcodes. They are used for various applications, including money transactions at funerals, billboard product advertisements, document management, etc.
    a. **Structure and Features**
        i. **Structure:** The QR Code is a 2-dimensional barcode consisting of black squares arranged on a white background within a square grid. It is scannable using a camera, such as those on smartphones or QR code readers.
        ii. **Capacity:** QR Codes can store 4,296 alphanumeric characters or 7,089 numeric characters.
        iii. **Error Correction:** QR Codes have built-in error correction, allowing up to 30% of the code to be damaged or obscured while still being readable.
        iv. **Variants:** QR Codes come in several versions, each capable of storing different amounts of data. The most common version is version 2, which can store 25x25 data modules.
    b. **Steps to Create QR Codes**
        i. First, you need to copy the data (e.g., URL, text, contact information) you want to use to create the QR code.
        ii. Launch a QR code generator on your preferred device that supports the QR code generator application.
        iii. Paste the copied data into the input area of the application
        iv. Click on generate (if required) and wait for the application to create the QR code image.
        v. Download it or screenshot it for your use.
    c. **Some applications of QR Codes**
        i. QR codes on packaging can link to product details or promotional offers.
        ii. Tickets with QR codes can link to event details and entry verification.
        iii. QR codes help track products through different stages of production and distribution.
        iv. QR codes are used for quick and secure payment processing.
        v. QR codes on documents can link to additional resources or digital copies.

LEARNING TASKS

1. Learners' scan QR codes using available scanning devices provided by the teacher.
2. Learners share their experience with using different QR codes and where they may have used them before, aside from the classroom QR code activity.
3. Learners watch a video demonstration of how to use QR code generators.
4. Learners, in their mixed-ability groups, create QR codes for any purpose they choose and exchange them with other groups, to scan their QR codes.
5. Learners discuss the process of creating a QR code with the whole class.

PEDAGOGICAL EXEMPLARS

1. **Explorational Learning**

The teacher provides learners with QR codes (this QR code is a resource for the next pedagogy) to scan using available scanning devices.

The teacher encourages the learners to share their experiences using QR codes and where they may have seen or used them before.

2. **Presentation and discussion**
   a. The teacher presents learners with a video demonstrating using QR code generators.
   b. The teacher guides learners in their mixed-ability groups to create QR codes for their chosen purpose.
   c. The teacher leads the discussion by asking the learners to share the process they used in creating the QR codes with the whole class and provides clarification feedback on the learners' contributions (if needed).

KEY ASSESSMENT

DoK Level 1: Recall

1. What does "QR" stand for in QR codes?
2. List two input devices used to scan QR codes.
3. What is the maximum number of numeric characters stored in a QR code?

DoK Level 2: Skills/Concept Application

1. Explain the error correction feature of QR codes and its importance.
2. Describe the steps to creating a QR code using an online generator.

DoK Level 3. Strategic thinking

1. Compare and contrast QR codes and traditional barcodes in retail settings.
2. Considering the impact on inventory management, customer experience, and overall efficiency in retail settings, which technology is more advantageous: QR codes or traditional barcodes? Discuss the strengths and weaknesses of each, and evaluate which system offers greater benefits for modern retail operations.

DoK Level 4. Expanded Thinking

Create a map of your school indicating your favourite food joint on campus, post it on your Facebook wall, and use the link to create a QR code to share with your friends, recommending the food joint.

HINT

This week's recommended mode of assessment is debate. You may use DoK level 3 question 2 as a sample. Refer to the Teacher Assessment Manual and Toolkit for additional information about how to administer this assessment mode.

WEEK 6

Learning Indicator: Explain the types of software and their functions

FOCAL AREA: TYPES OF SOFTWARE AND THEIR FUNCTIONS

1. **Major categories of software and their differences**
   a. **System Software:** Software designed to manage and control the hardware components of a computer and provide a platform for running application software. (E.g. Operating systems (Windows, macOS, Linux), device drivers, firmware, and utilities.). They facilitate the basic functions of a computer, such as executing applications and managing hardware resources.
   b. **Application Software:** Software designed to help users perform specific tasks or applications. (e.g. Microsoft Office Suite, web browsers (Chrome, Firefox), Adobe Photoshop). They are used to perform various tasks, such as word processing, web browsing, and data management.
2. **Functions of the system software**
   a. Operating Systems (OS) manage hardware resources, provide user interfaces, and serve as a platform for running applications. The operating system comprises various components: process management, I/O device management, file management, network management, main memory management, secondary storage management, security management, and a command interpreter system.
   b. Device drivers facilitate communication between the operating system and hardware devices.
   c. Firmware, embedded software in hardware devices, providing low-level control for device-specific functions.
   d. Utilities are system management tools that perform maintenance tasks, such as antivirus programs and disk cleanup tools.
3. **Functions of application software**
   a. Productivity software enhances efficiency in tasks like document creation, spreadsheets, and presentations (e.g., Microsoft Office).
   b. Web browsers enable users to access and navigate the Internet (e.g., Google Chrome, Mozilla Firefox and Safari).
   c. Media Players allow users to play audio and video files (e.g., VLC Media Player).
   d. Graphics software allows users to create and edit images and designs (e.g., Adobe Photoshop).

LEARNING TASKS

1. Learners differentiate between system and application software.
2. Learners explore different systems and application software and their features.

PEDAGOGICAL EXEMPLARS

1. **Interactive Presentation and Group Discussion:** The teacher gives an interactive presentation introducing the system and application software. The presentation must incorporate multimedia elements (video/images), real-world examples, and analogies to differentiate between the categories.
2. **Group work**: Use group work to enable collaborative research and brainstorming to discuss the use of system software and application software in real-world scenarios. Each group will be assigned a category (system or application software) to discuss.

KEY ASSESSMENT

DoK Level 1: Recall

1. Define system software and application software.
2. List three examples of system software and three examples of application software.

DoK Level 2: Skills/Concept Applications

1. Explain the primary purpose of an operating system.
2. Describe the role of device drivers in system software.

DoK Level 3: Strategic Thinking

1. Compare and contrast the functions of system utilities and application software.
2. Evaluate the need for both system and application software in a computer system.

DoK Level 4: Extended Thinking

1. Analyse, using real-world examples, the impact of a missing or malfunctioning device driver on a computer system's overall functionality.
2. Discuss the issues arising from the lack of compatibility of new application software with an existing system.

HINT

*This week's recommended mode of assessment is **mid-semester examination.** Refer to the Appendix C for a Table of Specification to guide you to construct your test items.*

SECTION 2 REVIEW

The creation and usage of barcodes and QR codes, as well as hardware and software functionality, are all covered in this section under computer architecture and organisation for learners. They comprehensively understand input and output devices by looking at different software packages and hardware elements. To promote peer collaboration and practical application, pedagogical methods include presentations, discussions, research-based learning, and project-based learning. A comprehensive evaluation of learners' comprehension and skill development is ensured by using Depth of Knowledge (DOK) assessments.

APPENDIX C: MID SEMESTER EXAMINATION

Structure

This mid semester examination covers focal areas from weeks 1-5 taking into consideration the DOK levels. The test should include:

Section A- Multiple Choice (20 questions answer all)

Resources

1. Computers
2. Answer booklets
3. Learner Material
4. Computing Teacher Manual
5. Teacher Assessment Manual and Toolkit

Sample Question (Multiple Choice)

1. Which of the following expressions will evaluate to True?

   Given that A = True, B = False, and C = True

   A. A AND B OR C

   B. B A OR B) AND NOT C

   C. NOT (A AND C) OR B

   D. A AND (NOT B OR C)

How to Administer

1. Administer constructed MCQs in a controlled environment to prevent cheating.
2. Provide clear directions/instructions to learners
3. Emphasise the importance of planning, drafting, revising and finalising the work.
4. Provide physical prompts or visual prompts to the learner who may require assistance
5. Provide support and accommodations for learners with special needs, such as extra time or a quiet testing environment.

Feedback

1. Provide targeted feedback to learners on each question.
2. Encourage learners to reflect on their strengths and areas they need to improve

Table of Specification

Week	Focal Area	Type of Question	DoK Level				Total
			1	2	3	4	
1	Understanding logic operations and expressions	Multiple Choice	1	2	1	–	4
2	Applying Boolean expression	Multiple Choice	1	2	1	–	4
3	Explanation of the Arithmetic Operations, Implementation of arithmetic operations in Python	Multiple Choice	2	2	1	–	5
4	Demonstrate creating and utilising barcodes and QR codes using input and output devices, such as scanners, digital cameras, keyboards, microphones, mice, 2D/3D scanners, and touch-screen sensors for control and monitoring.	Multiple Choice	1	1	2	–	4
5	QR Codes (Quick Response Codes)	Multiple Choice	1	1	1	–	3
	Total		6	8	6	–	20

SECTION 3: NETWORK SYSTEMS & TOPOLOGIES

STRAND: COMPUTER ARCHITECTURE & ORGANISATION

Sub-strand: Data Communication and Network Systems

Learning Outcome: *Apply skills and knowledge to design and demonstrate simple network topologies*

Content Standard: *Demonstrate knowledge and understanding of the components of a network*

INTRODUCTION AND SECTION SUMMARY

In this Computer Architecture & Organisation section, specifically focusing on Data Communication and Network Systems, learners will apply their skills and knowledge to design and demonstrate simple network topologies. The content standard emphasises understanding and demonstrating the components of a network. Learners will explore network topologies, their practical applications, and how they facilitate data communication. This foundation will enable learners to grasp how network systems are structured and function, preparing them for more advanced studies and practical applications in network design and management.

The weeks covered by the section are:

Week 7: Design and explain 2 network systems.

Week 8: Design and explain 2 network systems.

Week 9: Demonstrate how a network system can connect to the internet.

Week 10: Demonstrate how a network system can connect to the internet.

Week 11: Demonstrate how a network system can connect to the internet.

PEDAGOGICAL EXEMPLAR SUMMARY

This section adopted pedagogical approaches such as the flipped classroom. It also used video tutorials and online resources on local area networks (LANs). Collaborative group projects to design a LAN under teacher guidance were also adopted. Direct instructional activities and hands-on projects were also used in this section.

ASSESSMENT SUMMARY

This section uses Depth of Knowledge (DOK) to improve the quality, validity, and fairness of assessments while also promoting critical thinking and deeper learning among learners.

To ensure that all learner's abilities are evaluated, teachers are expected to use the weekly tasks and assessment questions to administer formative and summative tests on the material covered in "Network Systems & Topologies." This will allow them to identify areas for improvement and skills and use that information to plan lessons for the future.

It is important to note that the test items in this manual only serve as a guide for the teacher to establish learners' understanding of the lessons taught. They do not restrict the teacher's ability to develop original questions.

WEEK 7

Learning Indicator: Design and explain 2 network systems

FOCAL AREA: NETWORK SYSTEMS

1. **Designing a LAN**

Discuss the principles of good network design, including scalability, reliability, and security. When designing a Local Area Network (LAN), it is important to follow certain principles to ensure the network is efficient, scalable, secure and easy to manage.

a. **Scalability**: This refers to the ability to expand the network as the organisation grows without a complete redesign. Discuss best practices, like:

   i. The use of modular network devices can handle additional load.

   ii. Implement hierarchical design models like the three-layer model (core, distribution, and access).

   iii. Plan for future growth in terms of users, devices, and services.

   iv. Incorporate redundancies like backup paths and devices to ensure network availability in case of failures.

   v. Use redundant links and devices (e.g., dual power supplies, backup switches, and routers).

b. **Flexibility:** the ability to support a variety of devices and applications.

   i. Use standard protocols and interfaces.

   ii. Design for both wired and wireless access.

   iii. Ensure compatibility with diverse devices (e.g., computers, phones, and IoT devices).

c. **Performance:** ensuring the network can handle expected data loads and provide low latency.

   i. Use high-speed capacity links and switches (e.g., Gigabit Ethernet, 10 Gigabit Ethernet).

   ii. Optimise network traffic with Quality of Service (QoS) policies (e.g. Bandwidth allocation, Priority levels, Latency requirements and so on).

d. **Security:** protecting the network from unauthorised access and threats.

   i. Implement Firewalls: Configuring and understanding the use of firewalls.

   ii. Antivirus and Anti-Malware Software: Using and updating security software.

   iii. Use strong authentication mechanisms and regular software updates.

Planning: These include assessing requirements and selecting appropriate components and topologies. LAN (Local Area Network) planning involves several key steps to ensure that the network efficiently meets the organisation's current and future requirements. Here are the steps involved in LAN planning:

Define requirements and scope

Gather Requirements: Identify the number of users, devices, required applications, and coverage area.

Design Network Topology: To improve performance and security logically, select an appropriate network topology (e.g., star, bus, ring, mesh) based on scalability, redundancy, cost-effectiveness, and segment.

Select network equipment and connection media

Hardware Selection: Choose network devices (e.g., switches, routers, firewalls, and wireless access points) based on performance requirements, scalability, and budget constraints.

Connection Media: Determine the type of connection media to be used (e.g., wired or wireless). If wired, decide cabling type (e.g., STP, UTP, Coaxial, fibre-optic) and infrastructure (e.g., structured cabling systems) needed to support network connectivity.

Address IP Addressing and Subnetting

IP Addressing Scheme: Design an IP addressing scheme that accommodates current and future growth, taking into consideration segmentation (subnetting)

Plan for Network Security

Security Measures: Implement security measures such as firewalls, Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS), access control mechanisms, and encryption protocols to protect against unauthorised access and threats.

LEARNING TASKS

1. Learners recap computer networks they learnt in their first year.
2. Learners design Local Area Networks.
3. Explain network concepts and LAN designs.

PEDAGOGICAL EXEMPLARS

1. **Flipped Classroom with Online Resources:** Provide preparatory materials like videos and online tutorials on local area networks for learners to complete at home. some materials can be obtained at

Other materials can be obtained from

Learners will use class time to clarify concepts and work on group projects to design a LAN under the teacher's guidance.

2. **Direct instruction**
   a. Use multimedia presentations and videos to explain the nature of LAN and its layout, including where the various components are in the networks.
   b. Show videos, images, or other interactive tutorials to explain how to use a diagram-creating tool like Microsoft Visio to design layouts for real-world examples of wired LAN.
3. **Collaborative Project-Based Learning**: Engage learners in diverse groups to do hands-on projects where they design a Local Area Network (LAN) for their school's computer laboratory. Create mixed-ability groups for learners to design a LAN for their school's computer laboratory.

KEY ASSESSMENT

DoK Level 1: Recall

1. What is a Local Area Network (LAN)?
2. List three common components used in a LAN.

DoK Level 2: Skills/concept Application

1. Explain the primary purpose of a switch in a LAN.
2. Describe the difference between a wired and a wireless LAN.

DoK Level 3: Strategic Thinking

1. Design a simple LAN for a small office with 10 computers, including the necessary hardware components. Provide a diagram and an explanation of how the components are connected.
2. Analyse the impact of adding a switch to the existing LAN setup in a small office with 10 computers. How would it affect network performance?

DoK Level 4: Extended Thinking

1. Evaluate the security implications of implementing a wireless LAN (WLAN) in the same small office with 10 computers. Propose strategies to mitigate potential security risks.
2. Design a LAN for a larger organisation with multiple departments. Each department requires a separate network segment for security and performance reasons. Describe the design, including subnetting and necessary hardware components.
3. Design a poster of a simple LAN for a small office with 10 computers, including the necessary hardware components. Provide an explanation of how the components are connected.

HINT

This week's recommended mode of assessment is poster presentation, using Dok level 4 question 3. Refer to the Teacher Assessment Manual and Toolkit page 144 for additional information about how to administer this assessment mode.

WEEK 8

Learning Indicator: Design and explain 2 network systems

FOCAL AREA: NETWORK SYSTEMS

1. Understanding the Characteristics of CANs (Campus Area Networks, MANs (Metropolitan Area Networks), and WANs (Wide Area Networks).
   a. **Characteristics of CANs, MANs, and WANs**

      This will enable learners to identify the appropriate network type for a specified context:

      i. CANs typically have high-speed connectivity, often using ethernet or fibre optics. They connect multiple local area networks (LANs) that span a limited geographic area and are managed by a single organisation.

      ii. MANs have intermediate speeds: faster than WAN but slower than LAN. They connect multiple LANs within a metropolitan area and could be managed by one or multiple organisations.

      iii. WANs have varied speeds and often rely on leased lines and satellite links. They connect multiple MANs, CANs, and LANs, often involving multiple service providers and organisations covering a large geographic area, such as a country.

2. **Components of CAN/MAN/WAN design**
   a. The teacher discusses components and infrastructure elements for CAN/MAN/WAN design. The aim is to enable learners to identify appropriate devices to be used in designing a CAN/MAN/WAN:

      i. Routers direct data packets between networks, determining the best path for data to travel. They are used for interconnecting LANs within CANs, MANs, and WANs.

      ii. Switches connect individual devices within a single network, forwarding data to specific devices based on MAC addresses. Their role is to manage data flow within LANs, contributing to the overall performance of CANs and MANs.

      iii. Modems convert digital data into analogue signals for transmission over phone lines or other analogue media and vice versa. They are necessary for connecting to WANs, particularly for internet access via telephone lines.

      iv. Multiplexers combine multiple signals into one for transmission over a single medium and then separate them at the destination. They optimise the use of available bandwidth in WANs and MANs.

      v. Leased lines are dedicated, private communication lines provided by a telecom carrier for consistent, high-speed connectivity. WANs commonly use them for reliable, high-speed connections between distant locations.

   b. Teachers should discuss the security considerations for CAN/MAN/WAN Design

      i. Encryption protects data by converting it into an unreadable format that is only decrypted by authorised parties. It ensures data confidentiality during transmission across computer networks.

ii. Authentication aims to verify the identity of users and devices attempting to access the network. It helps prevent unauthorised access and ensures that only legitimate users can access network resources.

iii. Access control specifies policies and procedures for granting or restricting user access to network resources. By limiting access based on user roles and permissions, it helps maintain data integrity and confidentiality.

Teachers can access additional resources from

LEARNING TASKS

1. Learners form small groups to work on network design projects. Learners explain network concepts to their peers, reinforcing their understanding and helping others.
2. Learners present their designs, explaining their choices and the rationale behind their network architecture.

PEDAGOGICAL EXEMPLARS

1. **Direct instruction**
   a. Use multimedia presentations and videos to explain the nature of CANs, MANs, and WANs and their layout, including where the various components are located in the networks. Show videos/ images or other interactive tutorials to explain how to use a diagram-creating tool like Microsoft Visio to design layouts for real-world examples of CAN, MAN, or WAN.
   b. Introduce learners to network simulation tools like Cisco Packet Tracer or GNS3.
   c. Teachers can access the Cisco Networking Academy tools by scanning the following QR code:

2. **Collaborative Project-Based Learning**
   a. Engage learners in diverse groups to do hands-on projects where they design network systems for hypothetical or real-world scenarios.
   b. Assign learners to groups and provide a project brief, such as designing a network for a school (CAN), a city library system (MAN), or an international bank (WAN).

KEY ASSESSMENT

The summative assessment questions below are only to serve as a guide for the teacher when creating questions to measure learners' comprehension of the focal area.

DoK Level 1: Recall

1. State one difference between a CAN and a MAN
2. State one difference between a CAN and WAN

DoK Level 2: Skills/Concept Applications

1. Describe the main components required to design a Campus Area Network (CAN).
2. Explain how a Metropolitan Area Network (MAN) differs from a Wide Area Network (WAN) in terms of its characteristics and typical use cases.

DoK Level 3: Strategic Thinking

1. Analyse the school network requirements in a given town and design a suitable Metropolitan Area Network (MAN). Justify your choice of components and layout.
2. Compare and contrast the security considerations for designing CAN, MAN, and WAN networks.
3. Outline how you will set up the essential components needed to set up a Campus Area Network (CAN) in your school.

DoK Level 4: Extended Thinking

Discuss an integrated network solution that combines CAN, MAN, and WAN for the Ghana Education Service with headquarters in Accra and regional and district offices in different cities and towns. Provide a detailed explanation of your design, including the choice of technologies, infrastructure, and security measures.

HINT

This week's recommended mode of assessment is test of practical knowledge (TPK), using Dok level 3 question 3 as a sample

WEEK 9

Learning Indicator: Demonstrate how a network system can connect to the internet

FOCAL AREA: NETWORK CONNECTIVITY

Network connectivity establishes a link or connection between two devices - whether the devices are entirely different or similar. Without connectivity, our private and professional lives would be completely different. Network connections and data transmission are all around us, even though we may not know it. This makes our lives better. What kinds of connections are out there, and how do they function? The following are explanations of the different types of connectivity.

Types of Network Connections and Their Characteristics

1. **Wired (Ethernet) Connections:** A wired connection uses cables (usually Ethernet cables) to connect devices to a network, such as a local area network (LAN).

   Characteristics

   a. **Speed:** Wired connections are typically faster and more reliable than wireless connections. Common speeds are 100 Mbps (megabits per second) to 1 Gbps (gigabits per second).

   b. **Stability:** Ethernet provides a stable connection with minimal interference and fewer connection drops.

   c. **Security:** Wired networks are generally more secure because a physical connection is required to access the network.

   d. **Range:** limited by the length of the Ethernet cable, usually up to 100 metres.

   e. **Installation:** Requires running cables through walls or along floors, which can be cumbersome.

2. **Wireless (WiFi) Connections:** WiFi uses radio waves to connect devices to a network without cables.

   Characteristics

   a. **Convenience:** Allows devices to connect to the network without physical cables, making it easy to move devices around.

   b. **Speed:** Modern WiFi can be quite fast, with speeds up to several hundred Mbps, though typically slower than wired connections.

   c. Range: WiFi signals are capable of covering a house or office, but walls and other obstructions can interfere. Range extenders can improve coverage.

   d. **Interference:** susceptible to interference from other electronic devices and neighbouring WiFi networks, which can affect performance.

   e. **Security:** Requires strong passwords and encryption (like WPA3) to protect the network from unauthorised access.

3. **Cellular Connections:** Cellular networks use cell towers to provide wireless internet access over long distances, typically for mobile devices like smartphones and tablets.

Characteristics

a. **Mobility:** Allows internet access on the go, wherever there is cellular coverage.

b. **Speed:** Speeds vary by generation:

   i. **3G:** Older, slower speeds (up to a few Mbps).

   ii. **4G (LTE):** Faster speeds, commonly 20-100 Mbps.

   iii. **5G:** The latest technology, offering speeds up to 1 Gbps or more, with low latency.

c. **Coverage:** Coverage depends on the availability of cell towers and can vary between urban and rural areas.

d. **Data Limits:** Cellular plans often have data limits or may slow down after a certain usage threshold.

e. **Security:** Generally secure but using a VPN (Virtual Private Network) can add an extra layer of security.

Role of Network Devices in Establishing and Managing Internet Connections

1. **Routers:** A router is a device that connects different networks and directs data between them. It is often used to connect a local network (like a home or school network) to the internet. Many routers include WiFi capabilities - WiFi Routers, allowing wireless devices to connect to the network.

   Role in Network Infrastructure

   a. **Connecting Networks:** Routers connect your local network to the internet, enabling communication between your devices and online services.

   b. **Directing Traffic:** They manage and direct data traffic, ensuring that information reaches the correct destination.

   c. **Security:** Routers often have built-in firewalls to protect the network from external threats.

2. **Switches:** A switch is a device that connects multiple devices within a local network, allowing them to communicate with each other efficiently.

   Role in Network Infrastructure

   a. **Connecting Devices:** Switches connect devices like computers, printers, and servers within the same network.

   b. **Data Management:** They manage data traffic within the local network, ensuring data is sent only to the intended device, improving efficiency and speed.

   c. **Expansion**: Switches allow for network expansion by providing additional ports for more devices.

3. **Access Points (APs):** An access point is a device that allows wireless devices to connect to a wired network using WiFi.

   Role in Network Infrastructure

   a. **Extending WiFi Coverage**: Access points extend the coverage of a WiFi network, ensuring that wireless devices can connect to the network in different areas of a building.

b. **Improving Connectivity:** They improve overall wireless connectivity by providing additional points for connecting devices, reducing congestion.

c. **Central Management:** Some advanced access points can be managed centrally, making it easier to control and monitor the network.

Internet Protocols (TCP/IP)

- TCP/IP is the first set of protocols used on the Internet. It Allows computers to communicate/ share resources across the network. Work on TCP/IP started in the 1970s. It was funded by the US Military- ARPA. The modern internet sits on top of the TCP/IP technology. TCP/ IP is used as a standard to bridge the gap between non-compatible platforms. All computers connected to the internet understand the TCP/IP.
- Transmission Control Protocol (TCP): controls data packet sequencing, error checking, and reassembly to provide dependable data transfer between devices.
- Internet Protocol (IP): handles the addressing and routing of data packets across networks, ensuring the required data reaches the correct destination.

Scheme in TCP/IP Networks

- IPv4 stands for Internet Protocol version 4. It's a 32-bit address space, represented in dotted decimal notation (e.g., 192.168.1.1).
- IPv6 stands for Internet Protocol version 6. It is a 128-bit address space represented in hexadecimal format (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334). It has the advantages of vast address space, better security features, and improved routing efficiency.

Importance of Internet Protocols

Internet protocols play a major role in communication over the Internet; for instance,

a. The HTTP (Hypertext Transfer Protocol) is the standard protocol for web communication. Facilitating the transfer of web pages and resources over the internet.

b. The HTTPS (Hypertext Transfer Protocol Secures) also ensures secure data transfer between web browsers and servers using SSL/TLS encryption.

c. It also translates domain names (e.g., https://ges.gov.gh/) into IP addresses for navigating the internet and accessing websites.

d. Again, it also automatically assigns IP addresses to devices on a network to ensure network management and device configuration.

Authentication and Security Mechanisms

Authentication and security mechanisms are important for maintaining secure internet connections. Below are some key mechanisms used.

a. **Protected Access (WPA/WPA2 WiFi):** This secures wireless networks by encrypting data transmission and it protects WiFi networks from unauthorised access.

b. **Virtual private networks, or VPNs**, offer safe, encrypted connections across open networks. It ensures privacy and security for remote access and internet browsing.

c. Data sent between web servers and clients is encrypted via **SSL/TLS (Secure Sockets Layer/ Transport Layer Security)**. It protects sensitive data and online transactions.

LEARNING TASKS

1. **ICT Integration:** Learners watch a presentation of how wired (ethernet) and wireless (WiFi, cellular) connections are connected to the internet.
2. **Collaborative Research and Presentation**
   a. Learners in their mixed-ability groups research and present on what routers, switches, and access points are and their roles in network infrastructure.
   b. Learners, in their groups, research what WPA/WPA2, VPN, SSL/TLS, IP and TCP are and share their findings with the class.
3. **Mind-mapping:** Learners, in their groups, create a mind map detailing the concept of network connectivity, supervised by the teacher.
4. **Class discussion:** Learners watch and discuss a video on the authentication and security mechanisms of IP.

PEDAGOGICAL EXEMPLARS

1. **Direction Instruction**
   a. The teacher presents a video or slides to learners on how wired (ethernet) and wireless (WiFi, cellular) connections are connected to the internet. Emphasising the key steps in the connection process.
   b. Using a visual aid, the teacher explains the layered architecture of TCP/IP (Application, Transport, Internet, Network Access).
2. **Research and Presentation**: The teacher assigns specific research tasks to each learner group to research, and they present their findings.
3. **Mind-mapping**: The teacher supervises learners in their mixed-ability groups to create a mind map detailing the concept of network connectivity.
4. **Class discussion**
   a. The teacher shows learners a video on authentication and security mechanisms and discusses the process.
   b. Alternatively, the teacher can set up a demonstration on WiFi, VPN and SSL/TLS security for learners to facilitate the discussion.

KEY ASSESSMENT

DoK Level 1: Recall

1. Define TCP/IP and its primary function.
2. List the main differences between IPv4 and IPv6 addresses.

DoK Level 2: Skills/concept Application

1. Discuss the importance of SSL/TLS in secure data transmission.
2. Discuss the main difference between HTTP and HTTPS

DoK Level 3: Strategic Thinking

Explain how VPNs provide security for remote connections.

DoK Level 4: Extended Thinking

1. Design a secure network setup for a small business, incorporating various internet protocols and security mechanisms.
2. In groups, design a secure network infrastructure for a small business, incorporating appropriate internet protocols and security mechanisms.

HINT

This week's recommended mode of assessment is project-based, using Dok level 4 question 2 as a sample. Refer to the Teacher Assessment Manual and Toolkit page 125 for additional information about how to administer this assessment mode.

.

WEEK 10

Learning Indicator: Demonstrate how a network system can connect to the internet

FOCAL AREA: NETWORK SYSTEM CONNECTIVITY

Configuration and Setup Process for Network Connections

1. **IP Addressing**: An IP (Internet Protocol) address is a unique identifier for each device on a network.

   Steps to Configure IP Addressing

   Configuring an IP address can be one of two methods, which are explained below.

   a. **Static IP Assignment**

      i. Manually assign a fixed IP address to a device.

      ii. Typically used for servers, printers, and network devices.

      Example for Windows

      - Open Network and Sharing Centre.
      - Click on "Change adapter settings."
      - Right-click on the network connection and select "Properties."
      - Select "Internet Protocol Version 4 (TCP/IPv4)" and click "Properties."
      - Choose "Use the following IP address" and enter the IP address, subnet mask, and default gateway.

   b. **Dynamic IP Assignment**

      i. Use DHCP (Dynamic Host Configuration Protocol) to assign IP addresses automatically. It is commonly used for client devices like laptops and smartphones.

2. **DNS (Domain Name System)**: DNS translates human-readable domain names (e.g., www.example.com) into IP addresses that computers use to identify each other on the network.

   DNS Servers: Can be provided by ISPs, public DNS providers (e.g., Google DNS: 8.8.8.8), or configured locally.

   Steps to Configure DNS

   a. Access Network Settings

   b. Open the network settings on the device

   c. Navigate to the DNS settings section

   d. Enter DNS Server Addresses

   e. Enter the IP addresses of the preferred and alternate DNS servers.

   f. Example: preferred DNS server: 8.8.8.8, alternate DNS server: 8.8.4.4.

   g. Configuring Device Network Settings to Establish Internet Connectivity

3. **Steps to Configure WiFi on a Laptop/PC**

   Open Network Settings

   a. On Windows: Click on the WiFi icon in the taskbar and select "Network & Internet settings."

   b. On Mac: Click on the WiFi icon in the menu bar and select "Open Network Preferences" or "WiFi Settings"

   Select WiFi Network

   a. Click on "WiFi" or "Wireless" settings.

   b. Select the desired WiFi network from the list of available networks.

   Enter Password

   a. Enter the WiFi network password when prompted.

   b. Click "Connect" or "Join."

4. **Steps to Configure WiFi on a Smartphone**

   Open WiFi Settings

   a. On Android: Go to "Settings" > "Network & Internet" > "WiFi."

   b. On iPhone: Go to "Settings" > "WiFi."

   Select WiFi Network

   c. Choose the desired WiFi network from the list of available networks.

   Enter Password

   a. Enter the WiFi network password when prompted.

   b. Tap "Connect" or "Join."

5. **Setting Up Cellular Data Connections**

   Insert SIM Card

   Ensure a valid SIM card is inserted into the phone.

   Open Mobile Data Settings

   a. On Android: Go to "Settings" > "Network & Internet" > "Mobile Network."

   b. On iPhone: Go to "Settings" > "Cellular" or "Mobile Data."

   Enable Mobile Data:

   Toggle the "Mobile Data" or "Cellular Data" switch to "On."

   Configure APN (Access Point Name) Settings (if necessary)

   a. On Android: Go to "Mobile Network" > "Advanced" > "Access Point Names."

   b. On iPhone: Go to "Cellular" > "Cellular Data Network."

   c. Enter the APN settings provided by the cellular carrier

   d. Verify Connection: Ensure the device shows a successful Ethernet connection

   Open a web browser to check internet connectivity

6. **Steps to Configure Ethernet on a Laptop/PC**

Connect Ethernet Cable

a. Plug one end of the Ethernet cable into the device's Ethernet port.

b. Plug the other end into the router or switch.

Open Network Settings

a. On Windows: Right-click on the network icon in the taskbar and select "Open Network & Internet settings."

b. On Mac: Click on the Apple menu, select "System Preferences," and choose "Network."

Configure IP Settings (if necessary)

a. On Windows: Click on "Ethernet," then "Change adapter options." Right-click on the Ethernet connection, select "Properties," and choose "Internet Protocol Version 4 (TCP/IPv4)." Enter IP address, subnet mask, and default gateway if static IP is required

b. On Mac: Select "Ethernet," click "Advanced," and configure TCP/IP settings

LEARNING TASKS

1. Set up network connections on available devices, focusing on WiFi, cellular data, and Ethernet connections.
2. Configure IP addressing and DNS on available devices.

PEDAGOGICAL EXEMPLARS

1. **Demonstration:** The teacher demonstrates a configuration session. Emphasising on the IP addressing and DNS configuration
2. **Practical Session:** The teacher guides learners in the configuration process.
3. **Class Discussion:** The teacher facilitates an open discussion where learners share their experience and challenges in the configuration process.

KEY ASSESSMENT

DoK Level 1: Recall

1. Identify the steps involved in configuring IP addressing on a device

DoK Level 2: Skills/Concept Application

1. Explain how to set up a Wi-Fi connection on a laptop.
2. Discuss the differences you have identified when configuring Wi-Fi on a Laptop/PC compared to a smartphone device (pick either Android or iPhone)

Click on the link or scan the QR code to answer the questions

Marking Scheme

Correct answer (1 mark each)

1. Click on the Wi-Fi icon in the taskbar/system tray
2. Select the correct Wi-Fi network from the list
3. Enter the Wi-Fi password
4. Wait for the laptop to connect to the network
5. Manually select the network and enter the password again

DoK Level 3: Strategic Thinking

1. Discuss the most common issues encountered during the practical session and propose solutions to these issues.

HINT

*This week's recommended mode of assessment is **e-assessment**, using Dok level 2 question 3 as a sample. Refer to the Teacher Assessment Manual and Toolkit page 68 for additional information about how to administer this assessment mode.*

WEEK 11

Learning Indicator: Demonstrate how a network system can connect to the internet

1. **Internet Service Providers (ISPs)**

   Internet Service Providers (ISPs) are companies that provide individuals, businesses, and organisations with access to the internet.

   Functions of ISPs

   a. **Internet Access**: ISPs connect users to the internet by providing the necessary infrastructure, such as cables, routers, and data centres.

   b. **Bandwidth Management**: ISPs manage the bandwidth to ensure that data can flow smoothly between users and the internet.

   c. **Customer Support**: ISPs offer technical support to help users with connectivity issues and provide troubleshooting assistance.

   d. **Service Plans:** ISPs offer various internet service plans that differ in speed, data limits, and cost to cater to different user needs.

   e. **Security:** ISPs often provide security features like firewalls, anti-virus, and anti-spam filters to protect users from online threats.

2. **Types of Internet Access Technologies**

   a. **Dial-Up Internet**: An older technology that uses a phone line to connect to the internet.

   b. **Broadband Internet:** High-speed internet access that is always on and faster than dial-up.

   Types of Broadband Internet

   i. DSL (Digital Subscriber Line)

   ii. Cable Internet

   iii. Fibre-optic Internet

   iv. Satellite Internet

   Wireless Internet: Uses radio signals to provide internet access, often through cellular networks or WiFi hotspots.

   Wireless Internet Types

   3G (Third Generation)/ 4G (Fourth Generation)/ 5G (Fifth Generation): These technologies connect you to the internet through mobile devices like smartphones, tablets, and portable hotspots. They use radio waves to communicate with nearby cell towers, which are connected to the broader internet.

   Fixed Wireless provides broadband connectivity to a fixed location (e.g., a home or business) using wireless technology but here it emphasizes the “fixed” aspect, meaning the service is tied to a particular address and is not designed for mobility.

   Mobile and flexible are technologies that provide internet access while on the move, often through cellular networks. They provide a high-speed internet connection to a specific location using wireless technology,

3. Common Issues and Troubleshooting Techniques Related to Internet Connectivity.

	COMMON ISSUES	TROUBLESHOOTING TECHNIQUES
1	**Slow Internet Speed**: When Web pages load slowly, buffering during video streaming, or slow download/ upload speeds.	1. **Check Internet Plan**: Ensure your ISP plan supports the required speed. 2. **Restart Modem/Router**: Power cycle the modem and router to clear any temporary issues. 3. **Close Unnecessary Applications**: Close applications or tabs that may be consuming bandwidth or not needed at the moment. 4. **Update Firmware**: Ensure the router's firmware is up to date. 5. **Use Ethernet Connection**: For critical tasks, use a wired connection instead of WiFi to avoid interference.
2	**No Internet Connection**: When a client device is unable to access any websites or applications on the internet.	1. **Check Physical Connections**: Ensure all cables are properly connected and secure. 2. **Restart Devices**: Restart the modem, router, and computer. 3. **Check ISP Status**: Verify with your ISP if there is an outage in your area. 4. **Network Settings:** Check the network settings on your device to ensure it's set to obtain an IP address automatically.
3	**Intermittent Connectivity: When the** connection keeps dropping periodically.	1. **Signal Strength**: Move closer to the router or remove obstacles that might block the signal. 2. **Interference**: Identify and minimise interference from other devices (e.g., microwaves, cordless phones). 3. **Check for Overheating**: Ensure the router and modem are not overheating and are well-ventilated. 4. **Update Firmware:** Update the firmware of the router.
4	**DNS Errors:** When the DNS server is not responding. Giving the Error messages, inability to resolve domain names.	1. **Change DNS Server**: Change the DNS settings to use a public DNS server like Google (8.8.8.8, 8.8.4.4). 2. **Flush DNS Cache**: Clear the DNS cache on your device. 3. **Restart Router**: Restart the router to clear any temporary DNS issues. 4. **Windows**: Open Command Prompt and type ipconfig /flushdns. 5. **Mac**: Open Terminal and type sudo killall -HUP mDNSResponder.

5	**IP Address Conflicts:** When two devices on the same network are assigned the same IP address.	1. **Restart Devices**: Restart all devices connected to the network. 2. **Renew IP Address**: □. **Windows**: Open Command Prompt and type ipconfig /release followed by ipconfig /renew. □. **Mac**: Open Terminal and type sudo ipconfig set en0 BOOTP; sudo ipconfig set en0 DHCP. 3. **Assign Static IP**: Manually assign a unique static IP address to devices that frequently have conflicts.

LEARNING TASKS

1. Learners discuss what an Internet Service Provider (ISP) is and share which ISPs they are familiar with.
   a. Learners create a presentation explaining their assigned function of ISP, including real-world examples and how it benefits users.
   b. Groups take turns to present their findings to the class.
2. Learners watch pictures/videos of the types of Internet Access Technologies.
3. Learners troubleshoot a compiled list of common internet connectivity issues provided by the teacher.

PEDAGOGICAL EXEMPLARS

1. **Discussion:** The teacher begins the class with a discussion on what an Internet Service Provider (ISP) is and asks learners to share which ISPs they are familiar with.
2. **Collaborative Learning**
   a. The teacher divides learners into mixed-ability groups and assigns each group one of the functions of ISPs to research in detail.
   b. The teacher guides learners to create a simple presentation using flash-cards to explain their assigned function, including real-world examples and their benefits.
   c. The teacher gives each group a turn to present their findings to the class.
3. **ICT-Integration:** The teacher uses pictures/videos to introduce the different types of internet access technologies.
4. **Problem-Solving**: The teacher gives learners a set of common internet connectivity issues and tasks them to troubleshoot the issues.

KEY ASSESSMENT

DoK Level 1: Recall

1. What device is typically used to connect a home network to the internet?
2. What does the term ‘Broadband’ mean?

DoK Level 2: Skills/Concept Application

1. Briefly describe the steps involved in connecting a laptop computer to a Wi-Fi network.

2. How can an Ethernet cable be used to connect a desktop computer directly to the internet?
3. Describe the role of an Internet Service Provider (ISP) in providing internet access for a network

DoK Level 3: Strategic Thinking

Explain the role of an Internet Service Provider (ISP) in providing Internet access for a network.

DoK Level 4: Extended Thinking

Imagine you are setting up a network for a new business location. Analyse the factors to consider when choosing between a cable internet connection and a wireless internet connection for business needs.

HINT

This week's recommended mode of assessment is class exercise, using Dok level 2 question 3 as a sample.

SECTION 3 REVIEW

In this Computer Architecture & Organisation section, learners explored data communication and network systems, focusing on designing and demonstrating simple network topologies. Learners emphasised a network's components and examined various topologies and their applications. Pedagogical approaches included flipped classrooms with video tutorials, collaborative group projects, and direct instruction. Hands-on activities enhanced understanding and application of LAN, CAN, MAN and WAN designs. Assessments utilised Depth of Knowledge (DOK) levels to ensure comprehensive evaluation, critical thinking, and deeper learning, guiding future lesson planning and identifying areas for improvement. The assessment items serve as a guide for teachers to evaluate learners' understanding creatively.

SECTION 4: MAIN MEMORY & DATA STRUCTURES

STRAND: COMPUTATIONAL THINKING (PROGRAMMING LOGIC)

Sub-strand: Algorithm and Data Structure

Learning Outcome: *Apply skills in data storage processes to effectively manage and utilise data*

Content Standard: *Demonstrate knowledge and understanding of Data Storage*

HINT

- *The End of Semester Examination will be conducted in Week 12. Refer to **Appendix D** for a Table of Specification to guide you to set the questions. Set questions to cover all the indicators covered for at least weeks 1 to 11.*
- *Individual Project Work should be assigned to learners by the end of Week 13. Ensure that the project covers several learning indicators and spans over several weeks. Learners are expected to submit the individual project by week 20. Refer to **Appendix E** for further information on how to go about the individual project assessment.*

INTRODUCTION AND SECTION SUMMARY

In this section on Computational Thinking (Programming Logic), specifically focusing on Algorithms and Data Structures, learners will enhance their skills in data storage processes to manage and utilise data effectively, exploring various data storage methods, their applications, and how these methods enable efficient data management. This foundation will help learners grasp the complexities of data structures and algorithms, preparing them for advanced studies and practical applications in programming and data management.

The weeks covered by the section are:

Week 12: State and describe the functions of the Main Memory.

Week 13: Explain real-life applications of Data Structures.

PEDAGOGICAL EXEMPLAR SUMMARY

This section adopted pedagogical approaches such as demonstration, peer learning, and more to facilitate the teaching and learning of main memory and data structures.

ASSESSMENT SUMMARY

To encourage critical thinking and deeper learning among learners, this section uses Depth of Knowledge (DOK) to enhance the quality, validity, and fairness of assessments.

Note that the test items in this manual are intended solely to assist the teacher in determining whether or not the learners have understood the lessons that have been taught. They do not in any way limit the teacher's creativity when it comes to crafting questions.

WEEK 12

Learning Indicators

1. State and describe the functions of the Main Memory
2. Explain real-life applications of Data Structures

FOCAL AREA: MAIN MEMORY AND DATA STRUCTURES

1. **Main Memory (RAM—Random Access Memory)**

   Recap from year one, week 7, to ensure learners know what RAM is.

2. **Features of memory (RAM)**

   a. RAM is a computer's short-term memory, where the data and programs that the processor is currently using are stored.

      i. Capacity: RAM is measured in gigabytes (GB) for desktops, laptops and smartphones. The amount of RAM in a computer can significantly impact its performance, especially when handling multiple applications simultaneously. It commonly ranges from 2GB to 32GB or more.

      ii. Speed: Measured in MHz, it affects how fast data can be read and written to the RAM. The types of RAM (DRAM, SRAM) can affect the speed as well.

      iii. Volatility: RAM is volatile, meaning it loses its data when the computer is turned off.

      Types of RAM

      - DDR4,
      - DDR5 (Double Data Rate 4/5),
      - SDRAM (Synchronous Dynamic RAM).

3. **Introduction to Data Types**

   a. Integers are whole numbers that can be positive, negative, or zero. They are used for counting items such as the number of learners in a class or the score in a game.

      Example: -3, 0, 25.

   b. **Floats (Floating-Point Numbers):** Floats are numbers that have a decimal point. They can represent fractions or very large/small numbers. They are used in measuring quantities like temperature, height, or weight where precision is important.

      Example: 3.14, -0.001, 100.5

   c. **Characters:** Characters are single letters, digits, or symbols that are used to represent text. They are used for forming words, sentences, or any text-based information.

      Example: 'A', 'b', '3', '!', '#'

   d. **Pointers:** Pointers are variables that store the memory address of another variable. They "point" to the location of data in memory. They are used in managing dynamic memory, linking data structures like lists and trees.

      Example: If a variable stores the number 10, a pointer will store the address where 10 is located in memory.

4. **Introduction to Data Structures**: Data Structures are systematic ways of organising, managing, and storing data in a computer so that it can be accessed and modified efficiently. They are foundational to the field of computing and are used in almost every aspect of software development.

   Types of Data Structures

   Data structures can be categorized into sequential, tree-based, and graph-based types. Below is their breakdown:

5. **Sequential Data Structures:** Sequential data structures store data in a linear order, where each element is connected to its previous and next element.

   a. **Arrays:** An array is a collection of elements stored at contiguous memory locations. All elements are of the same type and are indexed.

      Example: **Array of Integers**: [5, 10, 15, 20]

   b. **Linked Lists:** A linked list is a collection of nodes where each node contains data and a reference (or link) to the next node in the sequence.

      Example: **Linked List of Integers**: 5 → 10 → 15 →20,

   c. **Stacks**: A stack is a collection of elements that follows the Last In, First Out (LIFO) principle. The last element added is the first one to be removed.

      Example:

      Stack of Books:[Book1, Book2, Book3]

      Add (Push) Book4: →[Book1, Book2, Book3, **Book4**]

      Remove (Pop) **Book4**:→ [Book1, Book2, Book3]

   d. **Queues:** A queue is a collection of elements that follows the First In, First Out (FIFO) principle. The first element added is the first one to be removed.

      Example:

      Queue of People: [Person1, Person2, Person3]

      Add (Enqueue) Person4: → [Person1, Person2, Person3, **Person4**]

      Remove (Dequeue) Person1:→ [Person2, Person3, Person4]

6. **Tree-based Data Structures:** Tree-based data structures organise data hierarchically, with a single root element and sub-elements (children). Each node can have zero or more child nodes.

   Components

   a. **Root**: The top node in a tree.

   b. **Node**: An individual element in the tree.

   c. **Child**: A node that descends from another node.

   d. **Parent**: A node that has one or more child nodes.

   e. **Leaf**: A node with no children.

   f. **Edge**: The connection between two nodes.

Example

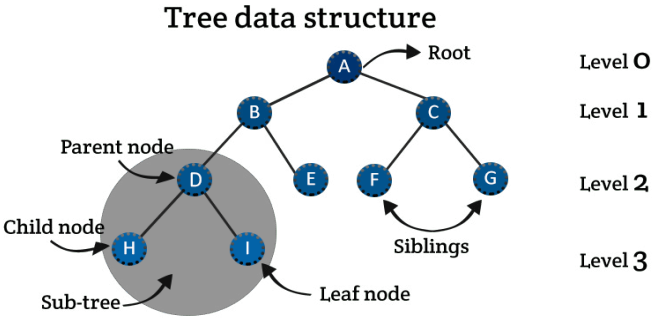


Figure 13.1: *Tree Data Structure*

Types of Trees

- **Binary Tree**: A tree where each node has at most two children.
- **Binary Search Tree (BST)**: A binary tree where the left child contains nodes with values less than the parent, and the right child contains nodes with values greater than the parent. It is used for searching and sorting and hierarchical data

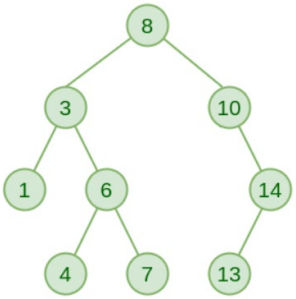


Figure 13.2: *Binary Search Tree*

7. **Graph-based Data Structures:** Graph-based data structures consist of nodes (vertices) and edges that connect pairs of nodes. Graphs can represent various complex relationships.

   a. **Graph:** A collection of nodes (vertices) connected by edges. Figure 13.3 is an example of a graph.

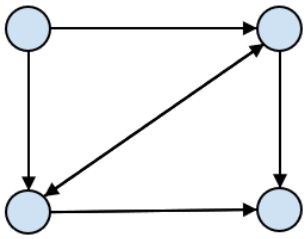


Figure 13.3: *Graph*

Components

- **Vertex (Node)**: An individual point in the graph.

Example: A, B, C, D

- **Edge**: The connection between two vertices.

Example: (A-B), (A-C), (B-D), (C-D)

Types

- Undirected Graph: Edges have no direction
- Directed Graph (Digraph): Edges have a direction
- Weighted Graph: Edges have weights

Usage: Modeling networks (social networks, communication networks), finding the shortest path.

LEARNING TASKS

1. **Recap Session**
   a. In an open discussion, learners explain what they remember about RAM
   b. Learners recap content from Year One, Week 7 on what RAM is and its role in a computer system
2. **Collaborative Learning**
   a. Learners, in groups, research one feature of RAM to explore (Capacity, Speed, Volatility).
   b. Each group then researches one of the types of RAM (DDR4, DDR5, SDRAM). Making use of diagrams where needed.
3. **Presentation and Debate**
   a. Each learner group presents their findings to the class.
   b. Learner groups have a structured debate on which type of RAM is most effective for the following specific tasks:
      i. gaming
      ii. server use
      iii. everyday computing
4. **Direct Instruction:** Learners pay attention to the explanation of what data structures are and why they are used in organising and managing data efficiently in computing.
5. **Collaborative Research**
   a. Learners research the differences between primitive and non-primitive data structures.
   b. In their groups, learners use flipcharts (if applicable) to present their findings on the differences between primitive and non-primitive data structures.
6. **Guided Practice**
   a. Learners watch a visual diagram of a tree and a graph structure demonstrated on the board by the teacher.
   b. Learners practice drawing trees and graphs facilitated by the teacher.

PEDAGOGICAL EXEMPLARS

1. **Recap Session**
   a. The teacher asks learners to explain what they remember about RAM.
   b. The teacher revises content on RAM from Year One, Week 7 with learners.
   c. The teacher clarifies any misconceptions or gaps in knowledge.
2. **Collaborative Learning**
   a. The teacher divides learners into mixed-ability groups and assigns each group one feature of RAM to explore focusing on ***Capacity, Speed, Volatility***.
   b. The teacher provides resources (textbooks, online articles) to aid learners' research.
3. **Presentation and Debate:** The teacher facilitates the flow of presentation and debate by timing the presentation and encouraging them to use evidence from their research to support their arguments.
4. **Direct Instruction:** Teacher explains what data structures are and why they are important for organising and managing data efficiently in computing.
5. **Collaborative Research**
   a. Teacher assigns learner groups to research the differences between primitive and non-primitive data structures.
   b. Teacher facilitates and guides the research process, providing resources and support as needed.
6. **Guided Practice**
   a. Teacher demonstrates visual diagrams of a tree and a graph structure on the board, explaining each component and its significance.
   b. The teacher guides learners to practice drawing trees and graphs.
   c. The teacher provides feedback to ensure learners understand the concepts correctly.

KEY ASSESSMENT

The summative assessment questions below are only to serve as a guide for the teacher when creating questions to measure learners' comprehension of the focal area.

DoK Level 1: Recall and Reproduction

1. What does RAM stand for?
2. Identify one type of RAM.
3. What happens to the data in RAM when the computer is turned off?
4. What is an integer? Provide an example.
5. Define an array and give an example.

DoK Level 2: Skills and Concepts

1. Explain the difference between DDR4 and DDR5 RAM.
2. How does the capacity of RAM affect a computer's performance?
3. Describe the role of RAM in a computer system.

4. Explain the difference between a stack and a queue.
5. Describe the purpose of pointers in programming.

DoK Level 3: Strategic Thinking

1. Compare and contrast the benefits of having 8GB of RAM versus 16GB of RAM for gaming.
2. Analyse why volatility is an important characteristic of RAM.
3. Evaluate the impact of RAM speed on running multiple applications simultaneously.

DoK Level 4: Extended Thinking

1. Formulate a debate argument for why a specific type of RAM (DDR4, DDR5, or SDRAM) is best suited for high-end gaming, incorporating technical specifications and real-world performance data.
2. Compare and contrast arrays and linked lists in terms of memory usage and performance.
3. Evaluate the advantages and disadvantages of using a binary search tree (BST) for data storage.

HINT

*The End of Semester Examination will be conducted in Week 12. Refer to **Appendix D** for a Table of Specification to guide you to set the questions. Set questions to cover all the indicators covered for at least weeks 1 to 11..*

WEEK 13

Learning Indicator: Explain real-life applications of Data Structures

FOCAL AREA: REAL-LIFE APPLICATIONS OF DATA STRUCTURES

1. **Hash-based Structures: Hash Tables and Maps:** Hash-based structures are an essential part of data organisation in computer science. They allow for efficient data retrieval, insertion, and deletion. The following are explanations of the different types.

   What are Hash-based Structures?

   Hash-based structures use a function to convert data into a unique index, called a hash code, which determines where the data should be stored or retrieved from in a data structure. They are used for quick data access, making operations like searching, inserting, and deleting very fast.

   a. **Hash Tables:** A hash table is a data structure that stores data in an array format but uses a hash function to calculate an index into an array of slots, from which the desired value can be found.

   b. **Components**

      i. **Hash Function**: A function that converts a given key into a specific index within an array.

      ii. **Buckets**: The slots or spaces in the array where data is stored.

      iii. **Keys**: The unique identifiers used to access values in the hash table.

      iv. **Values**: The actual data stored in the hash table.

      Example

      - **Keys and Values**: Imagine you have a list of learners and their IDs.
        - Key: Student ID
        - Value: Student Name

Student ID	Name
101	Raphael
102	Blessington
103	Gaddafi
104	Anokye
105	Millicent
106	Victor

Figure 13.1: *Student ID*

- **Collisions**: When two keys hash to the same index.
- **Handling Collisions**: Use methods like chaining (linking entries at the same index) or open addressing (finding the next available slot).

Example of Chaining

Index	Values
0	[Raphael, Mawusi]
1	[Blessington]
2	[Gaddafi, Victor, Anokye]
3	[Millicent]

Figure 13.2: *Chaining*

c. **Maps (Dictionaries):** A map, also known as a dictionary, is a collection of key-value pairs, where each key is unique and maps to a specific value.

Example: A **Phone Book**: Similar to the hash table example, a map can store phone numbers and names.

phone_book = {

"123-456-7890": "Alice",

"987-654-3210": "Bob",

"555-555-5555": "Charlie"}

2. **Real-life applications of data structures**

a. Arrays

Example: School Timetable

Application: Arrays can be used to create and manage school timetables. Each day of the week can be represented as an array, and each time slot within the day can be an element within that array. This allows for easy access and modification of the timetable.

b. Linked Lists

Example: Music Playlists

Application: Linked lists can be used to manage music playlists. Each song is a node in the list, and the next song is a pointer to the next node. This allows for easy addition, removal, and reordering of songs in the playlist.

c. Stacks

Example: Undo Function in Text Editors

Application: Stacks are used in text editors to implement the undo function. Each action (typing a letter, deleting a word) is pushed onto a stack. When the undo command is triggered, the most recent action is popped from the stack and reversed.

d. Queues

Example: Customer Service Lines

Application: Queues are used to manage customer service lines, whether physical or virtual (like call centres). Customers are served on a first-come, first-served basis. Each new customer is added to the end of the queue, and the customer at the front of the queue is served first.

e. Trees

Example: Family Tree

Application: Trees can be used to represent hierarchical structures like family trees. Each person is a node, and their children are connected as branches from their node, allowing for an organised representation of ancestry and relationships.

f. Graphs

Example: Social Networks

Application: Graphs can be used to model social networks where each person is a node, and each connection (friendship, follow) is an edge. This helps in understanding how people are connected and can be used to find the shortest path between two people, common friends, etc.

g. Hash Tables

Example: Dictionary

Application: Hash tables can be used to implement dictionaries where each word is a key and its definition is the value. This allows for fast look-up, insertion, and deletion of words.

h. Maps (dictionaries)

Example: Routing and Navigation Systems

Application: Maps are used to store geographical data and relationships between locations. Example: In a GPS navigation system, a map can store locations as keys and their coordinates or related metadata (like street names or landmarks) as values, enabling efficient route planning and location searches.

LEARNING TASKS

1. Learners explain data structures and their importance in computational thinking and problem-solving.
2. Learners watch a brief presentation on the main types of data structures (Arrays, Linked Lists, Stacks, Queues, Trees, Graphs).
3. Learners discuss how different data structures (Arrays, Linked Lists, Stacks, Queues, Trees, Graphs, Hash Tables) are used in real-life.
4. Learners research how arrays, linked lists, stacks, queues, trees, graphs, and hash tables are used in real-life.
5. Learners write five real-life applications of the data structures they researched. Explaining in their own words how these data structures are used.

PEDAGOGICAL EXEMPLARS

1. **Direct Instruction:** The teacher explains what data structures are and why they are important to computational thinking and problem solving.
2. **ICT-Integration:** The teacher shows learners a brief overview of the main types of data structures (Arrays, Linked Lists, Stacks, Queues, Trees, Graphs) using photos.
3. **Class Discussion:** The teacher facilitates a discussion on how Arrays, Linked Lists, Stacks, Queues, Trees, Graphs, Hash Tables are used in real-life.

4. **Research and presentation**
   a. The teacher guides learner groups to research how arrays, linked lists, stacks, queues, trees, graphs and hash tables are applicable to everyday life.
   b. The teacher tasks groups to write down 5 applications of their research and explain in their own words how any of the data structures are applicable.

KEY ASSESSMENT

DoK Level 1: Recall

1. List the seven main types of data structures.
2. Define what an array is.

DoK Level 2: Skills and concept Application

Match the data structures with their application to its application in the table below:

Array	A. Undo functionality in a text editor
Queue	B, Scheduling tasks in an operating system
Stack	C. Storing a list of daily temperatures

DoK Level 3: Strategic Thinking

Research how graphs are used in social network analysis and present your findings as homework.

DoK Level 4: Extended Thinking

Evaluate which data structures would be most effective in a web search engine and justify your choices.

HINT

*Individual Project Work should be assigned to learners by the end of Week 13. Ensure that the project covers several learning indicators and spans over several weeks. Learners are expected to submit the individual project by week 20. Refer to **Appendix E** for further information on how to go about the individual project assessment.*

SECTION REVIEW

This section builds foundational knowledge in data storage processes, focusing on main memory functions and real-life applications of data structures over two weeks. Through varied pedagogical approaches including demonstration and peer learning, learners develop skills to manage and utilise data effectively. Assessments employ Depth of Knowledge (DoK) methodology to promote critical thinking and deeper understanding, while teachers are encouraged to develop creative questions beyond the provided test items. This knowledge prepares learners for advanced programming studies and practical data management applications in computing environments.

APPENDIX D: END OF FIRST SEMESTER EXAMINATION

Structure

This end of semester examination covers focal areas from weeks 1-5. taking into consideration DOK levels. The test should include

1. Section A - Multiple Choice (30 questions answer all)
2. Section B - (5 Essay questions, 3 to be answered)
3. Test of Practical Knowledge (TPK) - (5 questions, answer 3)

Resources

1. Computer
2. Answer booklets
3. Learner Material
4. Teacher Manual
5. Teacher Assessment Manual and Toolkits

Sample Questions

Multiple Choice Question

1. Which of the following best describes a difference between a Campus Area Network (CAN) and a Metropolitan Area Network (MAN)?
   A. A CAN covers a large city, while a MAN is limited to a single building.
   B. A CAN typically covers a university campus, while a MAN covers an entire city.
   C. A CAN is used for international communication, while a MAN is used for local communication.
   D. A CAN is used in homes, while a MAN is used in businesses

Essay Type

Imagine you are setting up a network for a new business location. Analyse the factors to consider when choosing between a cable internet connection and a DSL internet connection for business needs

Test of Practical

Describe how to create barcodes. Assigned a specific barcode to your description

How to Administer (MCQ, ESSAY TYPE AND TPK)

1. Administer constructed MCQs in a controlled environment to prevent cheating.
2. Provide clear directions/instructions to learners
3. Emphasise the importance of planning, drafting, revising and finalising the work.
4. Provide physical prompts or visual prompts to the learner who may require assistance
5. Provide support and accommodations for learners with special needs, such as extra time or a quiet testing environment.

6. Encourage learners to manage time efficiently to complete tasks within allocated timeframes.
7. Encourage teamwork and effective communication if tasks involve group work.

Feedback

(MCQ, ESSAY TYPE AND TPK)

1. Score the essay consistently based on the established criteria to reduce subjectivity.
2. Give oral and written comments to the learner in the provision of feedback.
3. Encourage peer critiquing in the feedback process.
4. Give oral and written comments to the learner in the provision of feedback.
5. Encourage peer critiquing in the feedback process.
6. Give targeted feedback individually, in small groups and whole class

Encourage learners to reflect on their performance, review their work, and identify areas of improvement

Week	Focal Area	Type of Question	DoK Level				Total
			1	2	3	4	
1	Understanding Logic Operations and Expressions	Multiple Choice	2	2	1	–	5
		Essay	–	1	–	–	1
2	Applying Boolean expression	Multiple Choice	2	2	–	–	4
		Practical	–	–	1	–	1
3	Explanation of the Arithmetic Operations, Implementation of arithmetic operations in Python	Multiple Choice	2	2	–	–	4
4	Barcode	Multiple Choice	2	2	–	–	4
		Essay	–	–	1	–	1
		Test of Practical	–	–	1	–	1
5	QR Codes (Quick Response Codes)	Multiple Choice	1	1	1	–	3
6	Major categories of software and their differences, Functions of the system software, Functions of application software	Essay	–	–	1	–	1
7	Designing a LAN	Essay	–	1	–	–	1
		Test of Practical	–	1	–	–	1

8	Understanding the Characteristics of CANs (Campus Area Networks, MANs (Metropolitan Area Networks), and WANs (Wide Area Networks)	Multiple Choice	1	1	1	–	3
		Essay	1	–	–	–	1
9	Network connectivity	Multiple Choice	1	1	1	–	3
		Test of Practical	–	–	1	–	1
10	Configuration and Setup Process for Network Connections	Multiple Choice	–	1	1	–	2
		Test of Practical	–	1	1	–	2
11	Internet Service Providers (ISPs), Types of Internet Access Technologies, Wireless Internet	Essay	–	–	1	–	1
	Total		12	16	12	–	40

APPENDIX E: INDIVIDUAL PROJECT

Duration: Week 13 – Week 20

Task

Construct a logic circuit using only NAND gates that acts as an AND gate with two inputs (A and B). Provide the truth table for your circuit and explain how the combination of NAND gates produces the same output as an AND gate.

Rubric for Logic Circuit Construction

CRITERIA	*EXCELLENT (4)*	*GOOD (4)*	*SATISFACTORY (3)*	*NEEDS IMPROVEMENT (1)*
Critical Thinking	*Demonstrates understanding by using 4 of the skills:* *NAND inversion, 'Logical equivalence'.* *Truth table* *Circuit output*	*Demonstrates understanding by using 3 of the skills:* *NAND inversion, 'Logical equivalence'.* *Truth table* *Circuit output*	*Demonstrates understanding by using 2 of the skills:* *NAND inversion, 'Logical equivalence'.* *Truth table* *Circuit output*	*Demonstrates understanding by using 1 of the skills:* *NAND inversion, 'Logical equivalence'.* *Truth table* *Circuit output*
Problem-Solving	*Accurately applies logical reasoning to derive the circuit; identifies 4 steps:* *Truth table,* *Derive circuit, Invert output, logical steps*	*Accurately applies logical reasoning to derive the circuit; identifies any 3 steps:* *Truth table,* *Derive circuit, Invert output, logical steps*	*Accurately applies logical reasoning to derive the circuit; identifies any 2 steps:* *Truth table,* *Derive circuit, Invert output, logical steps*	*Accurately applies logical reasoning to derive the circuit; identifies any 1 step:* *Truth table,* *Derive circuit, Invert output, logical steps*
Communication	*Showing 4 of the skills e.g.* *1. Audible voice* *2. Keeping eye contact* *3. Pay attention to audience* *4. Engaging the audience with interaction* *5. Use of gesture*	*Showing 3 of the skills e.g.* *1. Audible voice* *2. Keeping eye contact* *3. Pay attention to audience* *4. Engaging the audience with interaction* *5. Use of gesture*	*Showing 2 of the skills e.g.* *1. Audible voice* *2. Keeping eye contact* *3. Pay attention to audience* *4. Engaging the audience with interaction* *5. Use of gesture*	*Showing 1 of the skills e.g.* *1. Audible voice* *2. Keeping eye contact* *3. Pay attention to audience* *4. Engaging the audience with interaction* *5. Use of gesture*

Total = 12 mark

SECTION 5: UTILISING PROGRAMMING LANGUAGE

STRAND: COMPUTATIONAL THINKING (PROGRAMMING LOGIC)

Sub-strand: App Development

Learning Outcome: *Demonstrate an understanding of fundamental concepts in text-based programming and the programming process and apply acquired skills to create new functions*

Content Standard: *Demonstrate knowledge and understanding of Computational Thinking (Programming)*

HINT

Conduct mid-semester assessment in Week 18. *Refer to* ***Appendix F*** *for the structure and table of specifications to guide you in setting the questions. Set questions to cover all the indicators covered for at least weeks 1 to 5.*

INTRODUCTION AND SECTION SUMMARY

In this section, we will explore the fundamentals of programming with Python by guiding learners through the development of a simple calculator application. The aim is to provide a hands-on learning experience that reinforces core programming concepts such as variables, control structures, functions and user input/output. By the end of this section, learners will have gained practical knowledge and skills in Python programming, enabling them to create functional programs that perform basic arithmetic operations.

The weeks covered by the section are:

Week 14: Utilise a programming language to develop a programme, such as a simple calculator (Basic Programming Concepts).

Week 15: Utilise a programming language to develop a programme, such as a simple calculator (Control Structures).

Week 16: Utilise a programming language to develop a programme, such as a simple calculator (Creating a Simple Calculator).

Week 17: Utilise simulation tools in Machine Learning to train a computer to perform specific actions or sets of actions.

Week 18: Utilise simulation tools in Machine Learning to train a computer to perform specific actions or sets of actions.

PEDAGOGICAL EXEMPLAR SUMMARY

This section adopted pedagogical approaches such as demonstration, peer learning, and more to facilitate the teaching and learning of main memory and data structures.

ASSESSMENT SUMMARY

To encourage critical thinking and deeper learning among learners, this section uses Depth of Knowledge (DoK) to enhance the quality, validity, and fairness of assessments.

Note that the test items in this manual are intended solely to assist the teacher in determining whether the learners have understood the lessons that have been taught. They do not in any way limit the teacher's creativity when it comes to crafting questions.

WEEK 14

Learning Indicator: Use a programming language to develop a programme, such as a simple calculator

FOCAL AREA: BASIC PROGRAMMING CONCEPTS

1. **Variables:** Variables are placeholders or containers in programming that hold data values. These values can be changed during the execution of a program. Think of a variable as a labelled box where you can store different items.

   Naming Conventions

   a. **Meaningful Names:** Choose names that reflect the variable's purpose or content. This makes your code easier to understand. For example, `age`, `total_price`, or `student_name` are descriptive names.

   b. **Case Sensitivity:** In Python, variable names are case-sensitive (this differs for different programming languages). This means `age`, `Age` and `AGE` would be considered different variables.

   c. **Start with a Letter or Underscore:** Variable names should start with a letter (a-z, A-Z) or an underscore (_). They cannot start with a number.

   d. **No Spaces or Special Characters:** Variable names cannot contain spaces or special characters like `@`, `#`, `$`, etc., except for underscores.

   e. **Use Underscores for Readability:** Use underscores to separate words in variable names for better readability, like `student_age` or `total_price`. This style is called snake case.

   f. **Avoid Reserved Keywords:** Do not use reserved keywords of the programming language as variable names (e.g., `if`, `else`, `while`).

2. **Data Types:** Data types define the kind of data a variable can hold, and this is important for ensuring code quality and correctness. Data types can be categorised into primitive data types and complex data types.

   a. **Primitive Data Types**

      i. **int (Integer):** Represents whole numbers without any decimal point. Integers are used for counting or indexing purposes. They are fundamental in operations like loops and conditions.

      Example: In Python: age = 16

      ii. **float (Floating Point):** Represents numbers with a fractional part. Floats are used when precision is needed, like in financial calculations, scientific measurements, and more.

      Example: In Python: price = 9.99

      iii. **char (Character):** Represents a single character. In Python, a single character is represented as a string (this is a sequence of characters, like letters, numbers, symbols, and spaces. Examples: `"Hello"`, `"123"`, `"Happy Birthday!"`) with a length of 1. Characters are used to represent letters, digits, punctuation marks, etc. They are important in text processing and user inputs.

Example: In Python: grade = 'A'

iv. **bool (Boolean):** Represents true or false values. Boolean values are used in decision-making processes within a program, especially in conditional statements.

Example: In Python: is_student = True

b. **Complex (non-primitive) Data Types**

i. **Array:** A collection of elements of the same, "homogeneous", data type, accessed by an index. Arrays are useful for storing collections of data like scores, names, or any list of items where you might want to perform similar operations on each element.

Example: In Python, lists can be used as arrays: scores = [90, 85, 88]

ii. **List:** Similar to arrays but more flexible. Lists can contain elements of different, "heterogeneous", data types and allow dynamic addition and removal of elements. Lists are versatile data structures for managing sequences of elements. They are widely used in various programming tasks.

Example: In Python: names = ["Gadaffi", "Raphael", "Milicent"]

iii. **Dictionary:** A collection of key-value pairs, where each key is unique. Dictionaries are efficient for looking up values associated with unique keys. They are particularly useful in scenarios where quick access to data is required.

Example: In Python: student_ages = {"Anokye": 16, "Raphael": 17, "Blessington": 14, "Pearl": 16}.

3. **Comments:** Comments are notes or explanations added to your code that are ignored by the computer when the program runs. They are used to make the code easier to understand for humans. Comments can explain what the code does, why it was written in a certain way, or provide any other information that might be helpful to someone reading the code, including your future self.

Why Use Comments?

Clarity: Helps others (and yourself) understand what the code is doing.

Documentation: Provides a way to document how the code works.

Debugging: Helps in finding and fixing errors by temporarily disabling code without deleting it.

Types of Comments

a. **Single-Line Comments**

These comments start with # and continue to the end of the line.

Example in Python

```
# This is a single-line comment
print("Hello, world!") # This prints a message
```

b. **Multi-Line Comments**

These comments span multiple lines.

Example in Python (using triple quotes):

```
""" This is a multi-line comment.
```

```
It can span multiple lines.
Used to explain more complex code.
"""
print("Hello, world!")
```

More examples of using comments

a. **Explaining Code**

```
# This variable stores the user's name
name = "Alice"
# Print a greeting message
print("Hello, " + name + "!")
```

b. **Documenting Steps**

```
# Step 1: Get user input
age = input("Enter your age: ")
# Step 2: Convert input to integer
age = int(age)
# Step 3: Check if the user is an adult
if age >= 18:
 print("You are an adult.")
else:
print("You are not an adult.")
```

c. **Debugging**

```
# Debugging: Comment out this line to test a different condition
# print("This line is temporarily disabled for debugging")
```

Functions

A function is a block of code designed to perform a specific task. It describes how to perform the particular task.

Why Use Functions?

1. Reusability: Write once, then use it many times. You can call a function multiple times without rewriting the code.
2. Organization: Break down complex problems into smaller, more manageable parts.
3. Readability: Makes your code easier to understand and maintain.
4. Avoid Repetition: Reduces redundancy by using functions to handle repetitive tasks.

Basic Structure of a Function

1. Function Definition: This is where you write the code for the function.
2. Function Call: This is where you use the function in your program.

Function Definition

How to define a simple function in Python:

```
def function_name(parameters):
# Code block
return value
```

Explanation

1. def: Keyword to define a function.
2. function_name: Name of the function.
3. parameters: Variables that the function takes as input (optional).
4. return: Keyword to return a value from the function (optional).

Example

Let us define a function that adds two numbers:

```
def add_numbers(a, b):
result = a + b
return result
```

Function Call

To use the function, you call it with arguments:

```
sum = add_numbers(3, 5)
print(sum) # Output: 8
```

1. **Arithmetic Operators:** Arithmetic operators are used to perform mathematical operations.
   a. **Addition (+):** Adds two numbers. Addition is fundamental for calculations involving the summation of values, such as total costs or scores.

      Example: In Python: sum = 5 + 3

   b. **Subtraction (-):** Subtracts one number from another. Subtraction is used to determine the difference between values, such as remaining balance or distance.

      Example: In Python: difference = 10 - 4

   c. **Multiplication (*):** Multiplies two numbers. Multiplication is essential for scaling values, like calculating area or total price.

      Example: In Python: product = 7 * 6

   d. **Division (/):** Divides one number by another. Division is used for distributing values, such as finding the average or splitting quantities.

      Example: In Python: quotient = 20 / 5 # quotient is 4.0

   e. **Modulus (%):** Returns the remainder of a division operation. The modulus operator is useful for finding remainders, like in cyclic processes or determining even/odd numbers.

      Example: In Python: remainder = 13 % 3 # remainder is 1

2. **Logical Operators:** Logical Operators are used to perform logical operations, often with Boolean values. (true / false)

a. **AND (and):** Returns True if both operands are true. The AND operator is used in conditions that require multiple criteria to be true, such as validating multiple user inputs.

Example: In Python: result = (5 > 3) and (8 > 6) # result is True

b. **OR (or):** Returns True if at least one operand is true. The OR operator is useful in conditions where only one of multiple criteria needs to be true, like checking multiple login methods.

Example: In Python: result = (5 > 3) or (8 < 6) # result is True

c. **NOT (not):** Inverts the Boolean value. The NOT operator is used to reverse the logic of a condition, useful in scenarios where you want to check for the absence of a condition.

Example: In Python: result = not (5 > 3) # result is False

3. **Comparison Operators:** Used to compare two values.

a. **Equal to (==):** Checks if two values are equal. The equality operator is essential in decision-making processes, like validating user input. **Example**: In Python: is_equal = (5 == 5) # is_equal is True

b. **Not equal to (!=):** Checks if two values are not equal. The inequality operator is used to enforce diversity or exclusion conditions, such as ensuring a unique username.

Example: In Python: is_not_equal = (5 != 4) # is_not_equal is True

c. **Greater than (>):** Checks if the left value is greater than the right value. The greater than operator is used in ranking or threshold-based conditions, like sorting scores.

Example: In Python: is_greater = (7 > 5) # is_greater is True

d. **Less than (<):** Checks if the left value is less than the right value. The less than operator is important in limit-based conditions, like filtering items under a certain price.

Example: In Python: is_less = (3 < 6) # is_less is True

e. **Greater than or equal to (>=):** Checks if the left value is greater than or equal to the right value. The greater than or equal to operator is used in inclusive threshold conditions, such as validating minimum requirements.

Example: In Python: is_greater_or_equal = (5 >= 5) # is_greater_or_equal is True

f. **Less than or equal to (<=):** Checks if the left value is less than or equal to the right value. The less than or equal to operator is used in inclusive limit conditions, like ensuring maximum capacity is not exceeded.

Example: In Python: is_less_or_equal = (4 <= 6) # is_less_or_equal is True

LEARNING TASKS

1. Learners practice how to declare variables and perform arithmetic and logic operations (individually, if applicable)
2. Learners write simple code and print to see the output**.**
3. Learners write a simple program making use of variables arithmetic and logic operations to complete the tasked challenge.

PEDAGOGICAL EXEMPLARS

1. **Demonstration and Practice**
   a. Teacher demonstrates to learners how to declare variables and perform arithmetic and logic operations.
   b. Teacher guides learners to write simple code and print to see the output.
2. **Code competition:** Teacher creates small exercises (e.g: write a program that takes two numbers as input and performs all arithmetic operations on them, displaying the results) where learners perform arithmetic, logical and comparison operations challenge in their mixed-ability groups.
3. **Reflective Learning**: Teacher facilitates a Q&A session, where learners share their understanding on the content learnt.

KEY ASSESSMENT

DoK Level 1: Recall

1. Explain the concept of a variable in programming.
2. What is the syntax to declare an integer variable in Python?

DoK Level 2: Skills and concepts applications

Write a program that takes a user's name and age as input and prints a greeting message along with their age.

DoK Level 3: Strategic thinking

Write a Python program that calculates the average of the three numbers 10, 20, and 30.

DoK Level 4: Extended Thinking

Design and implement an algorithm that sorts a list of learners' names in alphabetical order and then searches for a specific name using a binary search

HINT

*This week's recommended mode of assessment is **peer assessment**, using Dok level 2 question 1 as a sample.*

WEEK 15

Learning Indicator: Use a programming language to develop a programme, such as a simple calculator

FOCAL AREA: CONTROL STRUCTURES

1. **Conditional Statements:** Conditional statements are used to execute certain parts of code based on specific conditions. They allow the program to make decisions and follow different paths of execution.

   Types of Conditional Statements

   Branching: if, if-else, if-elif-else

   Branching in programming allows the code to make decisions and execute different paths based on certain conditions. It is like choosing different roads based on the traffic lights.

   a. **if Statement**

   Used to execute a block of code if a specified condition is true.

   Example:

   ```
   if temperature > 30:
   print("It's hot outside!")
   ```

   Explanation**:** If the temperature is greater than 30 degrees, the program will print "It's hot outside!"

   b. **if-else Statement**

   Used when there are two possible paths: one if the condition is true, and another if the condition is false.

   Example:

   ```
   if temperature > 30:
   print("It's hot outside!")
   else:
   print("It's not that hot.")
   ```

   Explanation: If the temperature is greater than 30 degrees, the program will print "It's hot outside!" Otherwise, it will print "It's not that hot."

   c. **if-elif-else Statement**

   Used when there are multiple conditions to check, one after another.

   Example:

   ```
   if temperature > 30:
   print("It's hot outside!")
   elif temperature > 20:
   print("It's warm outside!")
   else:
   print("It's cold outside!")
   ```

Explanation: If the temperature is greater than 30 degrees, it prints "It's hot outside!" If it's not greater than 30 but greater than 20, it prints "It's warm outside!" Otherwise, it prints "It's cold outside!"

Control Flow: Loops and Recursion

Control flow determines the order in which instructions are executed in a program. Loops and recursion are two primary control flow mechanisms.

1. **Loops**

   For Loop: Repeats a block of code a known number of times.

   Example,

```
for i in range(5):
 print(i)
```

   Explanation: This loop repeats 5 times and prints numbers from 0 to 4.

   While Loop: Repeats a block of code as long as a specified condition is true.

   Example,

```
count = 0
while count < 5:
print(count)
count += 1
```

   Explanation: This loop adds to the count variable each time it runs, it then prints numbers from 0 to 4 and will continue as long as the count is less than 5.

2. **Recursion**

   A technique where a function calls itself to solve a problem.

   Example

   Factorial Calculation

```
def factorial(n):
if n == 1:
return 1
else:
return n * factorial(n - 1)
print(factorial(5)) # Output: 120
```

   Explanation: The factorial function calls itself with n-1 until it reaches 1. The factorial of 5 is calculated as 5 * 4 * 3 * 2 * 1.

LEARNING TASKS

1. Learners revise the basic syntax and purpose of conditional statements, loops, recursion, and functions in Python.
2. Learners practice the use of if, if-else, and if-elif-else statements using python IDLE (if applicable). In the absence of computers, Learners should practice coding on paper.

3. Learners write a program that prints all even numbers between 1 and 50 using a for loop.
4. Learners write a program that asks the user for numbers until they enter 0 and then prints the sum of all entered numbers using a while loop.
5. Learners write a recursive function to find the sum of all elements in a list.
6. Learners create a function that takes a list of numbers as input and returns the largest number in the list.

PEDAGOGICAL EXEMPLARS

1. **Recap**: The teacher revises the basic syntax and purpose of conditional statements, loops, recursion and functions in Python.
2. **Demonstration**
   a. The teacher demonstrates the use of if, if-else, and if-elif-else statements.
   b. The teacher demonstrates the use of for and while loops
   c. The teacher explains and demonstrates recursion with a factorial example.
   d. The teacher explains and shows how to define and call functions.
3. **Reflective Learning**
   a. The teacher facilitates a discussion where learners share their experiences and challenges during the practical tasks.
   b. The teacher discusses common errors and misconceptions.

KEY ASSESSMENT

DoK Level 1: Recall

Write a program that takes two numbers as input and performs basic arithmetic operations (addition, subtraction, multiplication, division) on them, displaying the results.

DoK Level 2: Skills and concept application

1. Create a function that determines whether a given string is a palindrome (reads the same forwards and backwards).
2. Explain how conditional statements work in Python

DoK Level 3: Strategic Thinking

Write a recursive function to calculate the nth Fibonacci number.

DoK Level 4: Extended Thinking

Implement a for loop to print a multiplication table for numbers 1 to 10.

HINT

*This week's recommended mode of assessment is **class exercise**, using Dok level 2 question 2 as a sample.*

WEEK 16

Learning Indicator: Use a programming language to develop a programme, such as a simple calculator

FOCAL AREA: CREATING A SIMPLE CALCULATOR

This week, we will learn how to create a simple calculator using Python and the programming concepts we have learnt so far. A calculator is one of the fundamental tools in programming that performs arithmetic operations like addition, subtraction, multiplication, and division.

Steps in creating a simple calculator. Note that there are other alternative steps, the teacher can use.

1. **Prompt for User Input:** Use the `input()` function to prompt the user to enter the first number. Store this number in a variable (`num1`). Remember to convert the input to a float for numerical operations, like in the figure below (Figure 16.1).

```
# Creating a simple calculator in python
num1 = float(input("Enter the first number: "))
```

Figure 16.1: *step 1*

2. **Enter the Operator:** Prompt the user to enter an arithmetic operator (`+`, `-`, `*`, `/`). Store this operator in a variable (`operator`).

```
# Creating a simple calculator in python
num1 = float(input("Enter the first number: "))
operator = input("Enter an operator (+, -, *, /): ")
```

Figure 16.2: *step 2*

3. **Prompt for the Second Number:** Again, use the `input()` function to prompt the user to enter the second number. Store this number in a variable (`num2`), also converting it to a float.

```
# Creating a simple calculator in
python
num1 = float(input("Enter the first
number: "))
operator = input("Enter an operator
(+, -, *, /): ")
num2 = float(input("Enter the second
number: "))
```

Figure 16.3: *step 3*

4. **Perform Calculation Based on Operator:** Use conditional statements (`if`, `elif`, `else`) to perform the calculation based on the operator entered by the user. For each operator (`+`, `-`, `*`, `/`), compute the corresponding arithmetic operation.

```
# step 4
if operator == '+':
    result = num1 + num2
    print(result)
elif operator == '-':
    result = num1 - num2
    print(result)
elif operator == '*':
    result = num1 * num2
    print(result)
elif operator == '/':
    if num2 == 0:
            print("undefined: Division by zero is not possible")
    else:
        result = num1 / num2
        print(result)
else:
    print("Invalid operator entered.")
```

Figure 16.4: *Step 4*

5. **Display the Result:** Finally, use the `print()` function, as seen the figure above (Figure 16.4), to display the calculated result to the user. However, we can shorten the codes by calling the print function at the end of the code like in the figure below (Figure 16.5).

```
# step 4
if operator == '+':
    result = num1 + num2
elif operator == '-':
    result = num1 - num2
elif operator == '*':
    result = num1 * num2
elif operator == '/':
    if num2 == 0:
            print("undefined: Division by zero is not possible")
    else:
        result = num1 / num2
else:
    print("Invalid operator entered.")
#Step 5
print(f"Your result is: {result}")
```

Figure 16.5: *Step 5*

LEARNING TASKS

1. Learners revise basic programming concepts such as variables, functions, conditional statements, and loops under the guidance of the teacher.
2. Learners determine the features the calculator will have (e.g., addition, subtraction, multiplication, division)
3. Learners create a flowchart outlining the logic and flow of the calculator program
4. Learners, in their mixed-ability groups, create a calculator using the learnt programming concepts in Python.

PEDAGOGICAL EXEMPLARS

1. **Recap:** Teacher guides learners in revising basic programming concepts such as variables, functions, conditional statements, and loops to ensure they have a solid foundation.
2. **Class Project**
   a. Teacher facilitates a brainstorming session where learners determine the features their calculator will include (e.g., addition, subtraction, multiplication, division).
   b. Teacher assists learners in creating a flowchart outlining the logic and flow of the calculator program.
   c. Teacher oversees learners as they work in mixed-ability groups to develop a calculator using the programming concepts they have learned in Python.
3. **Presentation and Feedback**
   a. Teachers organises a session where learners present their developed calculators to the class.
   b. The teacher provides constructive feedback and guides learners on how to incorporate it for improvement.

KEY ASSESSMENT

DoK Level 1: Recall

1. Define a variable in Python and provide an example.
2. Write a Python code snippet that declares two variables and assigns them numeric values.

DoK Level 2: Skills and concepts Application

1. Explain how conditional statements work in Python.
2. Write a Python function that takes two numbers as input and prints which one is larger using an if-else statement.

DoK Level 3: Strategic Thinking

1. Describe the steps required to design a simple calculator program in Python.
2. Develop a flowchart for a calculator program that includes addition, subtraction, multiplication, and division functions. Implement the calculator using these functions and conditional statements in Python.
3. Click on the link below to play games on the roadmap

Blockly Games: https://blockly.games/puzzle?lang=en

Select the first game (Puzzle)

Follow the instructions to complete the game

Click on "Check Answers" to see your results

Move on to the next games and repeat the steps till you have exhausted the games

Share your experience with the class

DoK Level 4: Extended Thinking

1. Analyse how error handling can be incorporated into a calculator program to manage invalid inputs.
2. Extend the basic calculator program to handle division by zero errors and non-numeric inputs. Provide a detailed explanation of how you implemented these errors handling features and test your program with various inputs to demonstrate its efficiencies.

HINT

*This week's recommended mode of assessment is **gamification**, using Dok level 3 question 3 as a sample. Refer to the Teacher Assessment Manual and Toolkit page 82 for information on how to use this assessment mode.*

WEEK 17

Learning Indicator: Use simulation tools in Machine Learning to train a computer to perform specific actions or sets of actions

FOCAL AREA: UNDERSTANDING MACHINE LEARNING CONCEPTS

Introduction to Machine Learning: Learning Like a Machine

Machine learning (ML) is a branch of artificial intelligence (AI) that allows computers to learn from data and improve their performance over time, all without explicit programming. Imagine a student who gets better at solving maths problems, the more they practise, the better they get at it. That is the core idea behind Machine Learning.

Real-world examples of ML in our daily lives include the following

- **Recommendation Systems**

  Ever notice how Netflix suggests movies you might like, or how YouTube recommends videos based on your watch history

  These systems use ML to analyse your past choices and learn your preferences, recommending content they think you will enjoy.

- **Voice Assistant**

  Chatting with Siri on your iPhone or asking Alexa to play music? (Imagine a person talking to a smart speaker)

  Voice assistants use ML to understand your speech, convert it to text, and then respond to your requests or questions based on what they've learned about your voice patterns and past interactions.

Figure 17.1: *Voice Assistant*

- **Image Recognition**

  Unlocking your phone with facial recognition or tagging photos with friends on social media.

  ML algorithms analyse images and videos, allowing them to recognize objects, faces, and even emotions.

Types of Machine Learning

There are three main types of machine learning, each with its own approach to learning from data. Below are the various types of Machine Learning with an in-depth explanation.

Supervised Learning

Think of a teacher guiding a student. In supervised learning, the model is trained on **labelled data**, meaning each data point has a corresponding label or answer. The model learns the relationship between the input data and the desired output, enabling it to make predictions on new, unseen data.

Examples

Image Classification - Identifying different types of fruits.

Predicting House Prices - Using features(labels) like size and location to estimate the price.

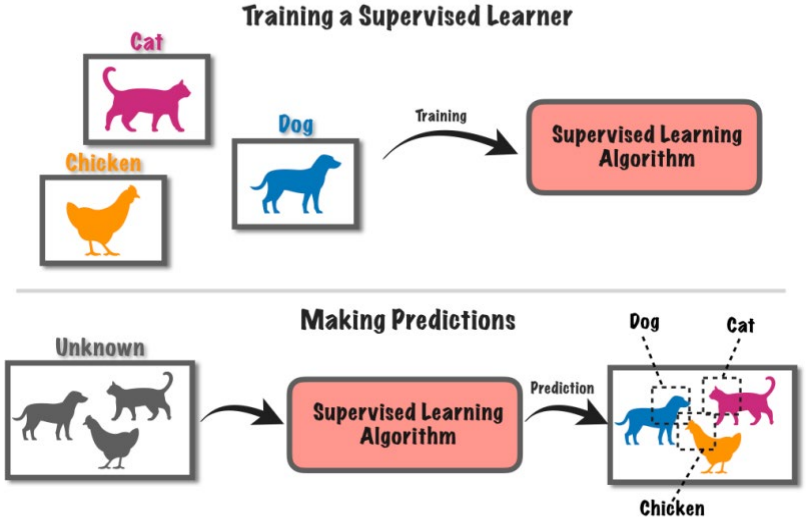


Figure 17.2: *Type of Machine Learning*

In Figure 17.2, the data were labelled before passing through the model; that is, each animal had its name tagged to it.

Unsupervised Learning

Unsupervised learning is a type of machine learning where the model is trained on data that has not been labelled or categorised. The goal is to identify relationships within the data without any prior knowledge of what the outputs should be. This contrasts with supervised learning, where the model is trained on a labelled dataset containing input-output pairs.

One of the most common tasks in unsupervised learning is called **clustering**. Clustering is the process of dividing data into groups, or clusters, based on their similarities. Each cluster contains data points that are similar to each other but different from data points in other clusters.

How does clustering work?

Clustering algorithms analyse the data points and calculate how similar or different they are from each other. This similarity is often measured using distance. Data points that are close together are considered similar and are grouped into the same cluster.

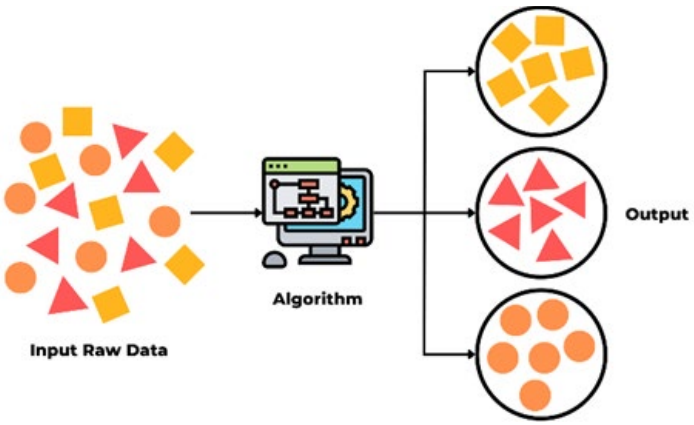


Figure 17.3: *Clustering*

Applications of Clustering

Clustering has a wide range of applications in various fields:

Customer Segmentation - Businesses can group customers based on their buying behaviour, demographics, or preferences to target marketing campaigns effectively.

Image Segmentation - Clustering can be used to identify different objects or regions within an image.

Document Clustering - Clustering helps organise large collections of documents into groups based on their content, making it easier to find relevant information.

Anomaly Detection - By identifying outliers that don't belong to any cluster, clustering can help detect unusual data points that might indicate errors or fraudulent activities.

Common Clustering Algorithms

There are many different clustering algorithms, each with its own strengths and weaknesses. Some common ones include:

K-means Clustering - This algorithm divides data into a predefined number of clusters (K).

Hierarchical Clustering - This algorithm creates a hierarchy of clusters, starting with individual data points and merging them based on similarity.

Reinforcement Learning

The model learns through trial and error, interacting with an environment and receiving rewards for desired actions and penalties for mistakes. Over time, the model learns the best course of action to achieve its goals.

Figure 17.4: *Reinforcement Learning*

Agent - The learner or decision-maker (in our analogy, the dog).

Environment—The external system with which the agent interacts (e.g., the yard or park where the dog is playing).

Action - What the agent can do (the dog running towards or away from the stick).

State/Observation - The current situation of the agent (the dog's position relative to the stick).

Reward—The feedback from the environment (e.g., a treat or praise when the dog catches the stick, or no reward if it fails).

Real-world examples of ML in our daily lives include the following:

- Essential in training self-driving cars to navigate roads, obey traffic laws and react to dynamic elements like pedestrians and other vehicles
- Reinforcement Learning is used to train robots and drones to navigate and perform tasks in dynamic and uncertain environments.
- Reinforcement Learning has been famously used to train agents to play and excel at games, both digital and physical.

Now that we understand the different types of machine learning and how they work, let's dive into the crucial first step of any machine learning project: data collection and preparation. Without high-quality data, even the most advanced algorithms will struggle to make accurate predictions, so it is essential to understand how to gather, clean, and prepare data effectively.

Data Collection and Preparation

The performance of a machine learning model heavily depends on the quality and quantity of the data used for training. High-quality data should be representative of the real-world scenarios the model will face. Below are the steps one has to go through for data collection to train a machine learning model, e.g. image data.

Define the Problem - Clearly define what you want to classify, such as types of fruits.

Collect Images

- Use a camera or smartphone to take pictures of the different types of fruits.
- Ensure a variety of images, considering different angles, lighting conditions, and backgrounds to make the model robust.

Label Images: Organise the images into folders, each representing a different class (e.g., one folder for apples, another for bananas).

Image Quality Tips

- Ensure images are clear and not blurry.
- Include a diverse range of images for each class to capture variations.
- Avoid repetitive images that could lead to overfitting.
- Now that we know about the types of machine learning models and the data collection preparations let us understand some of the libraries used to build these models. Let us first understand what a model and a library are before we start building a project.

Libraries are collections of pre-written code that help developers implement machine learning algorithms efficiently. They provide a wide range of functionalities that simplify the process of

developing, training, and deploying machine learning models. Below are the basic machine learning libraries learners must know.

NumPy: Think of NumPy as a tool that a mathematician might use to perform computations. It's a Python library designed to help you work with numbers and arrays (which are like lists of numbers). NumPy excels at performing various mathematical operations quickly and efficiently. For instance, if you need to add up all the numbers in a large list or carry out complex calculations, NumPy makes the process straightforward and fast.

Pandas: Pandas, on the other hand, is like an upgraded version of NumPy tailored for handling structured data, similar to tables in Excel. While NumPy focuses on numbers and arrays, Pandas is designed to help you organise and analyse data arranged in rows and columns. It is built on top of NumPy, leveraging its capabilities but adding a layer of powerful data manipulation features.

Scikit-Learn: Scikit-Learn, often referred to as Sklearn, is a powerful and easy-to-use open-source Python library for machine learning. It provides simple and efficient tools for data mining and data analysis, making it a valuable resource for building machine learning models.

A **machine learning (ML) model** is a mathematical representation that is trained on data to make predictions or decisions without being explicitly programmed to perform the task. The process involves feeding data to the model and allowing it to learn patterns and relationships within the data, which it can then apply to new, unseen data.

Key Components of a Machine Learning Model

Data—The input to a machine learning model, which can be structured data (tables, arrays) or unstructured data (images, text, audio).

Features - The attributes or variables in the data that are used to make predictions. Feature engineering involves selecting and transforming these features to improve the model's performance.

Algorithm - The method or procedure used to train the model. Examples include linear regression, decision trees, neural networks, and clustering algorithms.

Training - The process of using data to teach the model by adjusting its parameters to minimize error. This often involves splitting the data into training and validation sets.

Evaluation—Assessing the model's performance using metrics such as accuracy, precision, recall, F1 score, or mean squared error. This typically involves testing the model on a separate test dataset.

Prediction - Once trained, the model can be used to make predictions or decisions based on new input data.

LEARNING TASKS

1. Learners revise foundational concepts of machine learning (supervised learning, unsupervised learning, reinforcement learning, and neural networks).
2. Learners discuss the key components of Machine Learning and present their findings to their colleagues.
3. Learns go through the steps for data collection to train a machine learning model, e.g., image data.

PEDAGOGICAL EXEMPLARS

1. **Direct Instruction:** Teachers should explain foundational concepts of machine learning, including supervised learning, unsupervised learning, reinforcement learning, and neural networks.
2. **Collaborative Learning:** After the explanation, learners in their mixed-ability groupings should explore the navigation and features of popular simulation tools such as TensorFlow, PyTorch, Scikit-learn, or Keras.
3. **Demonstration:** Teachers should demonstrate data preprocessing techniques, including cleaning, normalisation, and preprocessing of datasets.

KEY ASSESSMENT

Level 1 (Recall and Reproduction)

1. What is machine learning? Answer: A field of computer science where algorithms learn from data without explicit programming
2. What is a simulation? Answer: An imitation of a real-world process or system

Level 2 (Skill-Concept and Applying Procedures): Identify an example of a machine learning task that could benefit from simulation-based training. (e.g., training a chatbot to have realistic conversations)

Level 3 (Strategic Thinking and Reasoning)

1. Explain why using simulation tools in machine learning can be advantageous compared to training with real-world data only.
2. Discuss the potential limitations of relying solely on simulation data for machine learning training. Answer: Simulations might not perfectly capture real-world complexities, leading to models that generalise poorly to new situations. (Biases might be introduced into the simulation design, leading to biased models.)

Level 4 (Extended Thinking and Reasoning): Imagine you are tasked with training a self-driving car to navigate different weather conditions. How can you design a simulation environment to address this challenge effectively? (Consider incorporating weather effects like rain, fog, and lighting conditions.) The simulation should be realistic enough to prepare the model for real-world variations.

HINT

*This week's recommended mode of assessment is **essay**, using Dok level 3 question 1as a sample. Refer to the Teacher Assessment Manual and Toolkit page 74 for information on how to use this assessment mode.*

WEEK 18

Learning Indicator: Use simulation tools in Machine Learning to train a computer to perform specific actions or sets of actions

FOCAL AREA: IMPLEMENTING MACHINE LEARNING AND ETHICAL CONSIDERATION

Let's consider a practical example of how to train a machine learning model to classify objects using a Google teachable machine simulation tool.

In this project, we built a model that can classify between a dog and a human. Below are steps to follow to achieve the desired model results.

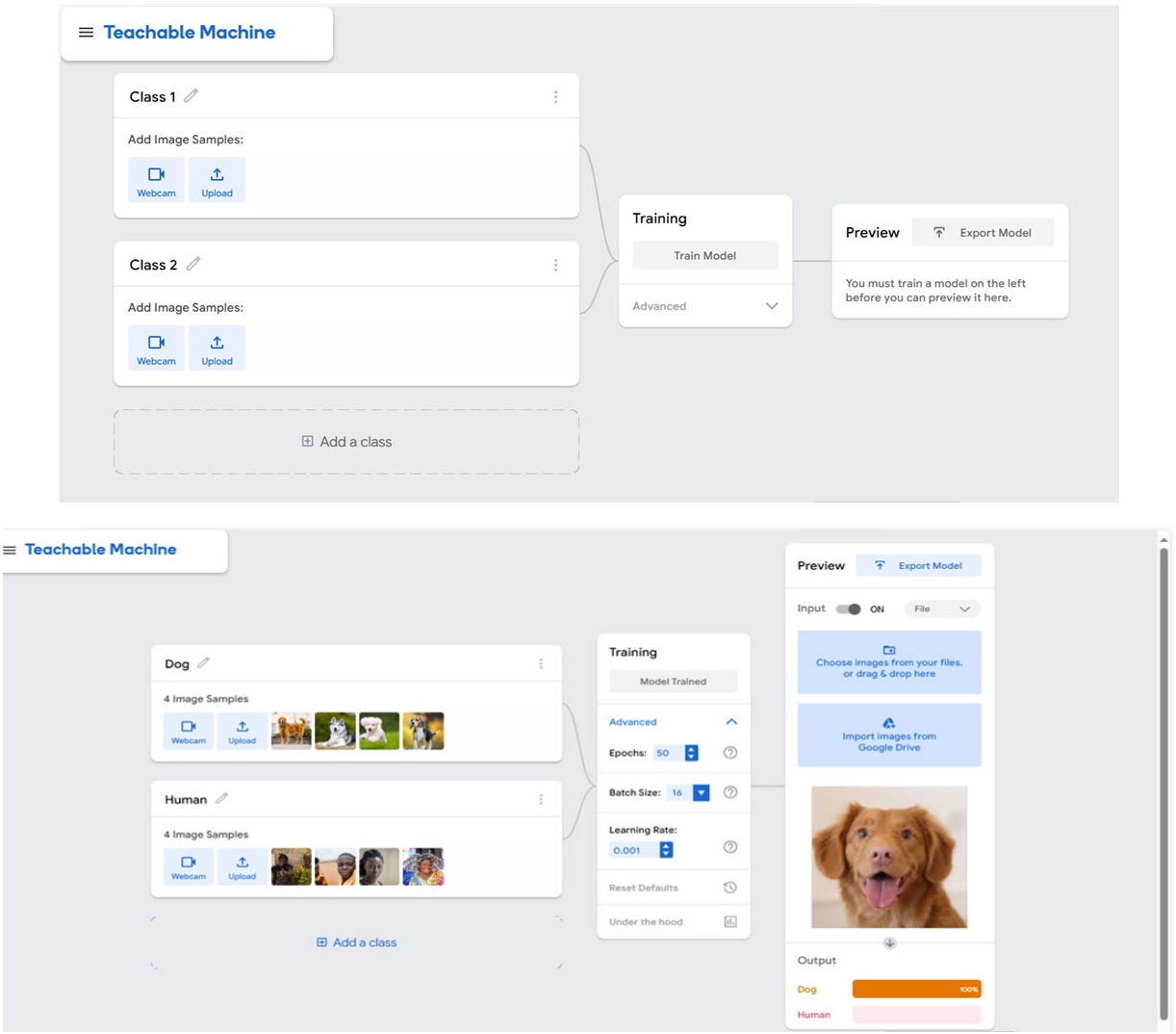
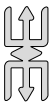


Figure 18:*1 Google Teachable Machine Learning*

- Go to the Teachable Machine website.
- Click on "Get Started" and choose "Image Project."
- Define classes like "Human" and "Dog."
- Capture images using your webcam or upload images from your computer for each class.

- Click “Train Model” and wait for the process to complete.
- Use new images to test the model’s predictions.



Note

There are other models we could have used for other tasks, such as

- Audio Project to train a model to recognise different sounds.
- Pose Project to train a model to recognise different poses.

Ethical Considerations in Machine Learning Models

Ethics refers to the principles that guide our decisions and actions to ensure they are fair, just, and respectful of others. In machine learning, ethics involves creating and using models in ways that are responsible and do not harm individuals or society.

Let us now take a look into the various ethical considerations to take into consideration when building a machine learning model and how to tackle it when they arise

Bias in training data happens when the data used to teach a machine learning model contains errors or unfair representations of certain groups.

How to Fix It

Use data from different sources, check for errors, and ensure that all groups are fairly represented.

Fairness means ensuring the machine learning model treats everyone equally, regardless of background.

How to Fix It

Use fairness checks, design the model to avoid bias, and involve diverse people in the development process.

Privacy involves protecting people’s personal information from misuse. For example, if a model uses student health records, it must keep that information confidential and secure.

How to Fix It

Anonymize data, use techniques to ensure privacy, and only collect necessary information.

Accountability means that developers are responsible for the model’s actions and must fix any problems. For example, if a model unfairly denied loans to certain groups, developers must identify and correct the issue.

How to Fix It

Test models thoroughly, follow guidelines and regulations, and have a way for people to report problems.

Let us consider the Potential Societal Impacts of machine learning models

Positive Impacts

1. Accurate models can help make better decisions in many areas, like predicting the weather or diagnosing diseases.
2. Machines can do routine tasks faster and more accurately, allowing humans to focus on complex problems.

Negative Impacts

1. If not designed carefully, models can worsen existing inequalities, making life harder for disadvantaged groups.
2. Using a lot of personal data can lead to privacy issues, like data breaches or misuse of information.
3. Complex models can be hard to understand, making it difficult to trust their decisions.

LEARNING TASKS

1. Learners revise foundational concepts and practical examples on how to train a machine learning model to classify objects using a Google teachable machine simulation tool.
2. Learners discuss the key components of building a simple machine learning model using an Iris flower dataset.
3. Learners build a machine learning model by considering the various ethical considerations and clearly stating how to solve them when they arise.

PEDAGOGICAL EXEMPLARS

1. **Talk for Learning**
   a. The teacher begins with an overview of data preprocessing, explaining its importance in ensuring the quality and suitability of datasets for training machine learning models.
   b. The teacher uses questioning and answering to tease out the role of data cleaning, normalisation, and pre-processing in improving model performance.
2. **Collaborative Learning:** Learners, in their mixed-ability groups, explore "data challenges and the impact of unprocessed data on machine learning outcomes." Using internet-connected devices and sharing their findings with the whole class.
3. **Demonstration:** The teacher conducts a live demonstration of data-cleaning techniques using a dataset with common issues. (showing how to handle missing values, remove duplicates, and correct inconsistencies.).

KEY ASSESSMENT

Level 1 (Recall and Reproduction)

1. What does it mean to "clean" data? Answer: Focus on removing errors/inconsistencies.
2. What does it mean to "normalise" data? Answer: Focus on scaling data to a common range.

Level 2 (Skill-Concept and Applying Procedures): How can you identify missing data in a spreadsheet?

Level 3 (Strategic Thinking and Reasoning)

1. Imagine you have a dataset with pet names. Some names are misspelt (e.g., "CAt" instead of "Cat"). How could you clean this data?
2. Why might normalising data be important when comparing different measurements (e.g., centimetres height vs. kilograms weight)?

HINT

*This week's recommended mode of assessment is **mid semester examination, refer** to Appendix F for more information on how to conduct this assessment.*

SECTION 5 REVIEW

In this section, learners developed a program, such as a simple calculator, using a programming language. They will grasp fundamental programming concepts such as variables, data types, operators, and control structures (e.g., loops, recursion, branching, including if statements and pattern matching).

In weeks 15 and 16, Learners improved their language proficiency by developing an understanding of the syntax and logic of a specific programming language suitable for the task, such as Python. Learners acquired syntax, data structures, and standard library skills.

In addition, Learners will develop skills in algorithmic thinking, testing and debugging, modular programming, user interface design, documentation and comments, version control and project management. This will be done in week 19.

Learners focused on utilising simulation tools in Machine Learning to train a computer to perform specific actions or sets of actions in weeks 17 and 18. They also acquired basic concepts of machine learning including supervised learning, unsupervised learning, reinforcement learning, and neural networks. They would also have learnt ethical considerations and potential societal impacts of their machine learning models (bias in training data, fairness, privacy, and accountability). Diverse teaching pedagogies and Webb's Depth of Knowledge (DOK) assessment levels ensure comprehensive understanding and critical thinking.

APPENDIX F: MID SEMESTER EXAMINATION

1. ***Structure***

This mid semester examination covers focal areas from weeks 13-17 taking into consideration DOK levels. The test should include

a) Section A - Multiple Choice (20 questions answer all)

2. ***Resources***

a. Computers
b. Answer booklets
c. Learner Material
d. Computing Teacher Manual
e. Teacher Assessment Manual and Toolkits

3. ***Sample Questions***

a) Multiple Choice Question

A queue is a data structure that follows the First-In-First-Out (FIFO) principle. Which of the following real-life scenarios best illustrates the use of a queue?

A. Filling water in a bucket where water at the bottom of the bucket is the first to be used.
B. Organising books on a shelf where the last book placed is the first to be removed.
C. Processing tasks in a printer where the most recent document is printed first.
D. Serving customers in a bank where the first person to arrive is the first to be served.

4. ***How to Administer***

a. Administer constructed MCQs in a controlled environment to prevent cheating.
b. Provide clear directions/instructions to learners
c. Emphasise the importance of planning, drafting, revising and finalising the work.
d. Provide physical prompts or visual prompts to the learner who may require assistance
e. Provide support and accommodations for learners with special needs, such as extra time or a quiet testing environment.

5. ***Feedback (MCQ)***

a) Score test items using the scoring rubric.
b) Provide constructive feedback to learners on each question.
c) Encourage learners to reflect on their strengths and areas they need to improve in
d) Evaluate the test item

6. *Table Of Specification*

Week	Focal Area	Type of Question	DoK Level				Total
			1	2	3	4	
13	Hash-based Structures, Real-life applications of data structures	Multiple Choice	2	2	1	-	5
14	Basic Programming Concepts	Multiple Choice	1	1	1	-	3
15	Control Structures	Multiple Choice	1	2	1	-	4
16	Creating a Simple Calculator	Multiple Choice	1	1	1	-	3
17	Understanding Machine Learning Concepts	Multiple Choice	1	2	2	-	5
	Total		6	8	6	-	20

SECTION 6: USE OF WEB PAGE EDITORS

STRAND: COMPUTATIONAL THINKING (PROGRAMMING LOGIC)

Sub-Strand: Web Technologies and Databases

Learning Outcome: *Apply acquired skills and knowledge to design and develop simple web pages*

Content Standard: *Demonstrate knowledge and understanding of Web development and databases*

HINT

*The Recommended Mode of Assessment for **Week 24** is End of Semester Examination. Refer to **Appendix G** for a Table of Specification to guide you to set the questions. Set questions to cover all the indicators covered for at least Weeks 13 to 24.*

INTRODUCTION AND SECTION SUMMARY

This section introduces learners to the use of web page editors, focusing on design elements, sections, text, images, GUI/UI/UX and multimedia integration. They will understand basic ideas like web page editors, HTML and Cascading Style Sheets (CSS), page structure and layout, formatting and typography, etc. The content is meant to give learners the skills they need to get the most out of programming logic and solve complex computational logic problems. By mastering these ideas, learners will be ready for further studies and career opportunities in computer science.

The section is interrelated with mathematics and electrical engineering. Teachers should intentionally emphasise the connections between these subjects to provide an understanding of how data storage and manipulation are integral to the broader field of computer science.

The weeks covered by the section are:

Week 19: Demonstrate using web page editors and design elements, including sections, text, images, GUI/UI/UX, and multimedia integration

Week 20: Demonstrate using web page editors and design elements, including sections, text, images, GUI/UI/UX, and multimedia integration

Week 21: Design an e-commerce web page for a selected company or organisation

Week 22: Design an e-commerce web page for a selected company or organisation

Week 23: Design an e-commerce web page for a selected company or organisation.

Week 24: Create a relational database model of a database

SUMMARY OF PEDAGOGICAL EXEMPLARS

This section employs diverse teaching strategies to ensure a comprehensive understanding, critical thinking, and peer learning. Some lessons begin with direct instruction to explain key concepts. The section adopts collaborative learning in mixed-ability groups to promote peer teaching and enhance understanding. Reflective discussions on fundamental HTML structural elements. Through experiential learning, learners identify critical points within programming logic. Q&A sessions led by the teacher are used to evaluate learners' understanding.

ASSESSMENT SUMMARY

This section uses Webb's depth of knowledge (DoK) to improve the quality, validity, and fairness of assessments while also encouraging learners to learn more deeply and think critically. The DoK levels ensure that tests correctly measure how well learners understand the content and give teachers the information they can use to plan and help learners understand the focal areas better.

To ensure that all learners' abilities are evaluated so that skills and areas for improvement can be identified and used to plan future lessons, teachers are to use the weekly tasks and assessment questions to give both formative and summative tests on the topics covered in "Computational Thinking (Programming Logic)."

It is important to note that the test items in this manual only serve as a guide for the teacher to establish learners' understanding of the lessons taught. They do not limit the teacher from exploring and creating innovative questions. Teachers are encouraged to refer to the Teachers Assessment Manual Toolkit (TAMK, NaCCA 2023) for in-depth knowledge of classroom assessment strategies.

WEEK 19

Learning Indicator: Demonstrate using web page editors and design elements, including sections, text, images, GUI/UI/UX, and multimedia integration

FOCAL AREAS: PAGE EDITORS AND DESIGN ELEMENTS

1. Introduction to Web Development
2. Basic HTML and CSS
3. Page Structure and Layout
4. Text Formatting and Typography

1.1 What is Web Development?

Web development is the process of creating websites and web applications that are accessible through the internet. It involves writing code, designing layouts, and managing content to build functional and interactive web pages. Web development can range from creating a simple static page of plain text to developing complex web-based applications, social network services, and e-commerce platforms.

Importance of Web Development

Web development is crucial in today's digital age because:

1. **Accessibility:** Websites make information and services accessible to people worldwide.
2. **Business Presence:** A website is essential for businesses to reach a wider audience and promote their products or services.
3. **Communication:** Websites and web applications facilitate communication through blogs, forums, and social media.
4. **Education:** Educational institutions use websites to provide resources and information to learners and educators.
5. **E-commerce:** Online stores and shopping platforms enable businesses to sell products directly to consumers over the internet.

1.2 Types of Websites (Static vs. Dynamic)

Static Websites

Static websites consist of fixed content. Each page is coded in HTML and displays the same information to every visitor.

Characteristics of Static Websites

1. Content does not change unless manually updated by the developer.
2. Easier and faster to create and host.
3. Generally cheaper to develop and maintain.
4. Suitable for small websites with limited content updates, such as portfolios or informational websites.

Example: A personal portfolio website that showcases an artist's work, with a few pages of content that rarely change.

Dynamic Websites

Dynamic websites are more complex and interactive. They use server-side technologies (like PHP, ASP.NET, or Node.js) and databases to generate content dynamically based on user interactions or other factors.

Characteristics of Dynamic Websites

1. Content can change based on user inputs, time, or other variables.
2. More interactive and engaging for users.
3. Can handle large amounts of data and frequent content updates.
4. Suitable for e-commerce sites, social networks, blogs, and other data-driven applications.

Example: An e-commerce website like Amazon, where the content (products, prices, recommendations) changes based on user interactions, inventory, and other dynamic factors.

Comparison

Feature	Static Websites	Dynamic Websites
Content Update	Manual	Automatic
Interactivity	Limited	High
Complexity	Simple	Complex
Cost	Lower	Higher
Development Time	Faster	Slower
Use Case	Portfolios, Informational Site	E-commerce, Social Media, Blogs

1.3 Role of Databases in Web Development

Databases

Databases store and manage data for web applications. They allow web developers to save, retrieve, and manipulate information as needed.

Types of Databases

Databases are broadly categorised into two types: SQL and NoSQL databases.

SQL Databases: Use structured query language (SQL) to manage data. Examples include MySQL, PostgreSQL, and SQLite.

Relational Model: Data is organised into tables with rows and columns. Tables can be linked using primary and foreign keys.

Queries: SQL commands to perform operations on the data, such as SELECT, INSERT, UPDATE, and DELETE.

Teachers can guide learners to practise the following query

```sql
SELECT * FROM users WHERE age > 18;
INSERT INTO users (name, age) VALUES ('John', 25);
UPDATE users SET age = 26 WHERE name = 'John';
DELETE FROM users WHERE age < 18;
```

NoSQL Databases: Use various data models, such as documents, key-value pairs, or graphs. Examples include MongoDB, CouchDB, and Redis.

Document Model: Data is stored in JSON-like documents. (Note: JSON means JavaScript Object Notation, which is a standard text-based format for representing structured data and is a lightweight way of storing and transferring data across the internet.)

Key-Value Pair Model: Data is stored as key-value pairs.

Graph Model: Data is stored as nodes and relationships.

Teachers are to guide learners to read further on CRUD (Create, Read, Update, Delete) Operations and Server-Side Languages such as PHP, ASP.NET or Node.js.

Understanding Web Page Editors

2.1 Introduction to Web Page Editors

Web page editors are software tools that help developers create, edit, and manage the content and design of websites. These editors provide a user-friendly interface that simplifies the web development process, making it accessible to beginners and efficient for experienced developers.

Purposes of Web Page Editors

1. **Simplify Web Development:** Web page editors provide tools and features that simplify creating and editing web pages.
2. **Streamline Workflow:** They streamline the workflow by integrating various web development tasks, such as coding, designing, and debugging, into a single platform.
3. **Enhance Productivity:** Web page editors enhance productivity and reduce development time by providing features like syntax highlighting, code completion, and drag-and-drop interfaces.
4. **Ensure Compatibility:** Web page editors often include tools for testing and ensuring compatibility across different browsers and devices.

Types of Web Page Editors

Web page editors can be broadly categorised into WYSIWYG (What You See Is What You Get) editors and code editors.

WYSIWYG Editors

WYSIWYG editors allow developers to create web pages visually without writing code. These editors provide a graphical interface where users can drag and drop elements, such as text, images, and buttons, onto a canvas. The editor generates the underlying HTML, CSS, and JavaScript code automatically.

Features

1. **Drag-and-Drop Interface:** Users can easily add and arrange elements on the page by dragging and dropping them.
2. **Real-Time Preview:** Changes made in the editor are immediately reflected in the preview, which shows how the final web page will look.
3. **Templates and Themes:** WYSIWYG editors often come with pre-designed templates and themes that can be customised to suit users' needs.
4. **Integrated Tools:** Many WYSIWYG editors include tools for managing images, multimedia, and other assets.
5. Examples:
6. **Adobe Dreamweaver is a** popular WYSIWYG editor that offers a wide range of features for both beginners and advanced users.
7. **WordPress is a** content management system (CMS) that includes a WYSIWYG editor for creating and managing website content.
8. **Wix and Squarespace:** These are online platforms that provide WYSIWYG editors for building and hosting websites.

Code Editors

Code editors are designed for developers who prefer to write their code themselves. They provide a text-based interface where users can write HTML, CSS, JavaScript, and other web development languages. Code editors offer advanced features that help developers write, edit, and manage code efficiently.

Features of code editors

1. **Syntax Highlighting:** Code editors highlight different code elements in different colours, making it easier to read and understand.
2. **Code Completion:** These editors suggest code completions and auto-complete code as the user types, reducing errors and speeding up development.
3. **Debugging Tools:** Code editors often include debugging tools that help identify and fix errors in the code.
4. **Version Control Integration:** Many code editors integrate with version control systems, such as Git, allowing developers to manage changes to their code. They also provide an audit trail of changes and rollback features in case of any issues that arise once code has been deployed.

Example code editors

1. **Visual Studio Code is a** popular code editor from Microsoft that supports a wide range of programming languages and extensions.
2. **Sublime Text:** A lightweight and highly customisable code editor known for its speed and efficiency.
3. **Atom is a**n open-source code editor developed by GitHub. It is known for its flexibility and wide range of community-developed packages.

2.2 Popular Web Page Editors

In this section, Teachers are to guide learners to research further on some of the most popular web page editors: Adobe Dreamweaver, WordPress, Wix, and Squarespace. These tools offer a range of features and interfaces designed to simplify the web development process.

Basic HTML and CSS
What is HTML?

HTML, or Hypertext Markup Language, is the standard language for creating and designing web pages. It provides the basic structure of a web page by using a system of tags and attributes to define elements and their content.

HTML is used to structure content on the web. It tells the web browser how to display text, images, and other media. HTML elements are the building blocks of web pages, defining everything from headings and paragraphs to links and images.

HTML is composed of elements enclosed in angle brackets (tags). Each element has an opening and closing tag, with the content placed between them. For example, the below code shows the use of the   tag which is used to create paragraphs on the webpage:

```'html
<p>This is a paragraph.</p>
```

The basic structure of an HTML document

An HTML document has a specific structure that includes several essential elements. Here is a breakdown of a basic HTML document:

1. **Document Type Declaration**

   Specifies the version of HTML being used. It is placed at the very top of the document.

   ```html
   <!DOCTYPE html>
   ```

2. **HTML Element**

   - The root element that contains all other elements on the page.

   ```html
   <html>
   # Elements that will be displayed on the webpage
   </html>
   ```

3. **Head Element**

   Contains meta-information about the document, such as the title and links to stylesheets.

   ```html
   <head>
       <title>Page Title</title>
   </head>
   ```

4. **Body Element**

   Contains the content of the web page that is displayed to the user.

   ```html
   <body>
       <h1>Welcome to My Web Page</h1> # this is a heading
       <p>This is a paragraph of text.</p> # this is a paragraph of text
   </body>
   ```

Common HTML Elements

HTML elements define various parts of a web page. Here are some common HTML elements and their uses:

1. **Headings**

   Headings are used to define titles and subtitles on a page. There are six levels of headings, from `<h1>` (largest) to `<h6>` (smallest).

   ```html
   <h1>Main Heading</h1>
   <h2>Subheading</h2>
   ```

2. **Paragraphs**

   Paragraphs are used to define blocks of text. They are enclosed in `<p>` tags.

   ```html
   <p>This is a paragraph of text. It can contain multiple sentences and will be displayed with space above and below it, also referred to as padding.</p>
   ```

3. **Links**

   Links are used to navigate to other web pages or resources. They are defined with the `<a>` tag and require an `href` attribute to specify the URL.

   ```html
   <a href="https://www.example.com">Visit Example.com</a> # the text between the <a> and </a> tags will be what is displayed to the user on the screen and it will be a clickable link
   ```

4. **Images**

   Images are embedded in a page using the `<img>` tag. The `src` attribute specifies the image file and the `alt` attribute provides alternative text for accessibility.

   ```html
   <img src="image.jpg" alt="Description of image"> # the alt text is used for accessibility and SEO reasons
   ```

5. **Lists**

   Lists can be ordered (numbered) or unordered (bulleted). Ordered lists use the `<ol>` tag, and unordered lists use the `<ul>` tag. List items are defined with the `<li>` tag.

```html
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
</ul>
```

6. **Tables**

   Tables are used to display data in a tabular format. They use the `<table>`, `<tr>`, `<th>`, and `<td>` tags to define the table structure.

```html
<table>
  <tr> # this starts the table row
<th>Header 1</th> # this defines the table header
<th>Header 2</th>
  </tr>
  <tr>
  <td>Data 1</td> # this defines the table data
  <td>Data 2</td>
  </tr>
</table>
```

Introduction to CSS

What is CSS?

CSS, or Cascading Style Sheets, is a language used to describe the presentation of a web page written in HTML. CSS controls the layout, colours, fonts, and overall appearance of web content. While HTML provides the structure of a web page, CSS handles how that structure looks to the users. This is what makes the webpages user-friendly.

CSS allows you to apply styles to HTML elements, making it possible to create visually appealing and well-designed web pages. It separates the content (HTML) from its presentation (CSS), promoting better organisation and maintenance of web projects.

Benefits of CSS

Consistency: Apply the same styles across multiple pages.

Flexibility: Easily change the appearance of your website without modifying HTML content.

Separation of Concerns: Keep the structure (HTML) and styling (CSS) separate for better organisation.

CSS Syntax and Selectors

CSS syntax consists of rules that define how HTML elements should be styled. Each rule is made up of a selector and a set of declarations. Here's the basic syntax:

p is a selector in CSS – pointing to the element to be styled. Note that this can be any element on the page, such as h1 or h2

colour is a property, and the colour would be set at the property value e.g. blue

text-align is a property, and the alignment would be set as the property value e.g. centerNote: Teachers are to guide learners in practising with examples. This must be done as a project in their mixed-ability groups.

3.3 Combining HTML and CSS

CSS can be used to style various HTML elements to enhance the visual presentation of a web page. Applying CSS to HTML elements is essential for creating well-designed and aesthetically pleasing web pages. Here's how to effectively combine HTML and CSS:

Using Inline CSS

Inline CSS is applied directly to an HTML element using the `style` attribute. This method is useful for quick, one-off styles but is less efficient for applying styles to multiple elements.

Example, showing how to set the font colour to blue and the size to 20px:

```html
<p style="color: blue; font-size: 20px;">This is a paragraph with inline CSS styling.</p>
```

NOTE

the spelling of color when writing CSS.

Using Internal CSS

Internal CSS is included in the `<style>` section of the HTML document's `<head>`. It is useful for styling elements within a single HTML document.

LEARNING TASKS

1. Learners revise foundational concepts and practical examples of designing a basic website, applying all the knowledge gained from this class.
2. Learners discuss the key components of Basic HTML and CSS. This should be done with examples and presented to their colleagues.

PEDAGOGICAL EXEMPLARS

1. **Explorative Learning**: The teacher will facilitate the lesson by guiding learners to explore popular web page editors such as Adobe Dreamweaver, WordPress, Wix, or Squarespace. This will include teaching them how to navigate the interface, create new pages, and manage existing content
2. **Collaborative Learning:** After the exploration, learners in their mixed-ability groups will choose a web page editor, create a new page, and manage some existing content. They will document their experience and share their findings with the class.

KEY ASSESSMENT

Level 1: (Recall and Reproduction)

1. What is the purpose of a web page editor?
2. What does HTML stand for?
3. What is the basic function of CSS in web design?

Level 2: (Skill/Concept)

1. How do you create a new webpage in a web page editor?
2. Can you explain the difference between inline and block-level elements in HTML?
3. What are some common CSS properties used to style text on a webpage?
4. How do you create a navigation bar using HTML and CSS?

Level 3: (Strategic Thinking)

1. Compare and contrast the features of different web page editors and explain which one you would choose for a specific project and why.
2. Analyse a webpage's source code and identify how HTML and CSS have been used to structure and style the content.
3. Evaluate the effectiveness of a webpage's layout and design, suggesting improvements based on HTML and CSS principles.
4. Design a responsive web page layout using media queries in CSS to accommodate different screen sizes.

Level 4: (Extended Thinking)

1. Develop a custom webpage editor tool with advanced features beyond basic HTML and CSS editing.
2. Create a comprehensive guide for beginners on best practices for writing clean and efficient HTML and CSS code.
3. Propose innovative ways to use HTML and CSS to enhance user experience on a webpage, such as animations or interactive elements.
4. Research and present a case study on the impact of using advanced CSS techniques, such as flexbox or grid layout, on webpage performance and user engagement.

HINT

*This week's recommended mode of assessment is **practical assessment**, using Dok level 3 question 4 as a sample.*

WEEK 20

Learning Indicator: Demonstrate using web page editors and design elements, including sections, text, images, GUI/UI/UX, and multimedia integration

FOCAL AREAS

1. Image Integration
2. GUI/UI/UX Principles
3. Multimedia Integration

1.1 Adding Images and Videos: Multimedia elements such as images and videos are crucial for enhancing web pages' visual appeal and engagement. In this section, you will learn how to incorporate these elements into your web pages using HTML.

1. **Using `<img>` Tag for Images:** The `<img>` tag in HTML is used to embed images on a web page. It is a self-closing tag, which means it doesn't require a closing tag. Here's how you can use it effectively:

   Basic Syntax

   ```html
   <img src="URL" alt="description" width="width" height="height">
   ```

   Attributes

   `src`: Specifies the URL of the image. This can be a relative path (e.g., `images/photo.jpg`) or an absolute URL (e.g., `https://example.com/photo.jpg`).

   `alt`: Provides alternative text for the image if it cannot be displayed. This is important for accessibility and SEO (search engine optimisation).

   `width` and `height`: Define the dimensions of the image in pixels.

2. **Embedding Videos with `<video>` and `<iframe>`**

   Videos can be integrated into web pages using either the `<video>` tag for self-hosted content or the `<iframe>` tag for embedded content from video-sharing platforms like YouTube and Vimeo.

   Using the `<video>` Tag: The `<video>` tag lets you include video files directly from your server. It supports various video formats like MP4, WebM, and Ogg.

   Basic Syntax

   ```html
   <video controls width="width" height="height">
       <source src="URL" type="video/format">
       Your browser does not support the video tag.
   </video>
   ```

Attributes

a. `controls`: Adds playback controls (play, pause, volume) to the video player.

b. `width` and `height`: Define the dimensions of the video player.

c. `<source>`: Specifies the video file's path and format.

1.2 Integrating Audio: Integrating audio into your web pages can enhance the user experience by providing sound effects, background music, or narrated content. In this section, you'll learn how to add audio elements using HTML and customize their controls.

1. Using `<audio>` Tag: The `<audio>` tag is used to embed audio files into your web page. It allows you to provide a variety of audio formats and include playback controls for users.

Basic Syntax

```html
<audio controls>
    <source src="URL" type="audio/format">
    </audio>
```

Attributes

a. `controls`: Adds playback controls (play, pause, volume) to the audio player.

b. `<source>`: Specifies the audio file's path and format (e.g., MP3, Ogg).

1.3 Interactive Multimedia: Interactive multimedia elements can enhance user engagement by providing dynamic and interactive experiences. In this section, you'll learn how to embed interactive content such as maps and slideshows and use plugins and widgets to extend the functionality of your web pages.

1.4 Best Practices for Multimedia: When integrating multimedia into your web pages, it is important to follow best practices to ensure your content loads efficiently, looks great, and is accessible to all users. This section covers key practices for optimising media and ensuring accessibility.

1. **Optimising Media for the Web:** Optimising multimedia ensures that your website performs well and provides a smooth experience for users, regardless of their device or internet connection.

a. **Image Optimisation**

Choosing the Right Format:

Joint Photographic Expert Group (JPEG): Good for photographs and images with many colours. Offers good compression with minimal loss in quality.

Portable Network Graphic (PNG): Best for transparent images or when you need high-quality graphics.

WebP: Provides high-quality compression and smaller file sizes but ensures browser compatibility.

Compressing Images: To compress images without sacrificing quality, use tools like [TinyPNG](https://tinypng.com) or [JPEG-Optimizer](https://www.jpeg-optimizer.com).

b. **Video Optimisation**

Choosing the Right Format

MP4: A widely supported format that offers good video quality and compression.

WebM and Ogg: Alternative formats that offer better compression but may have limited support.

Compressing Videos: Use tools like [HandBrake](https://handbrake.fr) or [Clipchamp] (https://clipchamp.com) to reduce file sizes while maintaining quality.

LEARNING TASKS

1. Learners listen to and interact with teachers and peers, take notes, and ask questions to clarify their understanding of HTML tags.
2. learners actively collaborate in their group to develop the sample website they have been assigned.

PEDAGOGICAL EXEMPLARS

1. **Interactive Lecture and Discussion:** The teacher will introduce learners to the basics of web design concepts using a multimedia presentation that includes videos and examples of well-designed websites. Discuss the functions and importance of sections, text, videos and images in web design and multimedia file formats. Use a video to demonstrate how HTML tags add text, videos, images, attributes, and alt text. Introduce learners to a web page editor like WordPress, Wix, or Google Sites.
2. **Class Discussion:** Engage learners in discussing their favourite websites and what design elements make them appealing.
3. **Hands on practice group project:** Assign learners to mixed-ability groups. Provide them with sample code to type into Microsoft Word or Notepad, save as an HTML file, and open in a browser to practice developing web pages and HTML tags.

KEY ASSESSMENT

Level 1 (Recall and Reproduction)

1. What are some common web page editors?

2. Identify some basic design elements used in web pages.

Level 2 (Skill, Concept and Applying Procedures)

1. Briefly describe the steps in creating a simple web page using a web editor.
2. Explain the purpose of using headings and subheadings on a web page.

Level 3 (Strategic Thinking and Reasoning)

1. How can the effective use of images and multimedia content enhance a web page's impact and engagement?
2. Explain the difference between GUI (Graphical User Interface), UI (User Interface), and UX (User Experience) in web design.
3. Discuss the importance of using sections to structure a web page and improve user experience (Sections can group related content and guide users visually).

Level 4 (Extended Thinking and Reasoning)

Imagine you are tasked with creating a website for a local business. How can you apply your understanding of web page design elements, accessibility and user experience (UX) principles to create a user-friendly and visually appealing website?

HINT

*This week's recommended mode of assessment is **critiquing**, using Dok level 3 question 3 as a sample. Refer to the Teacher Assessment Manual and Toolkit page 58 for information on how to use this assessment mode.*

WEEK 21

Learning Indicator: Design an e-commerce web page for a selected company or organisation

FOCAL AREAS

1. Responsive Design & Accessibility Considerations
2. Product Presentation and Catalogue
3. Shopping Cart and Checkout Process

Designing an E-commerce Web Page

Introduction to Responsive Design

Responsive design refers to designing and developing websites that adapt seamlessly to various screen sizes and devices. This ensures that users have an optimal viewing experience, whether using a desktop computer, tablet, or smartphone.

Importance of Responsive Design

1. **Improved User Experience:** Responsive design provides a consistent and user-friendly experience across all devices, which can lead to higher user satisfaction and engagement.
2. **Increased Mobile Traffic:** With the growing use of smartphones and tablets, a responsive design ensures that your website is accessible and functional on mobile devices, potentially increasing traffic.
3. **Cost-Effective:** Maintaining a single responsive website is more cost-effective than creating and managing separate versions for different devices.
4. **Search Engine Optimization (SEO) Benefits:** Search engines like Google favour responsive designs, which can improve your website's ranking in search results. Google recommends responsive web design as the best practice for mobile configuration.
5. **Future-Proofing:** Responsive design helps ensure your website remains functional and relevant as new devices and screen sizes emerge.

Key Principles of Responsive Design

1. **Fluid Grid Layouts:** Use relative units, like percentages instead of fixed units like pixels, to create flexible grid layouts. This allows elements to automatically resize and rearrange based on the screen size.

   Example: Instead of setting a div's width to 300px, set it to 50% of its container size.

2. **Flexible Images:** Images should scale and adapt to different screen sizes. Use CSS properties like `max-width: 100%;` to ensure images resize within their containing elements.

   Example: `img { max-width: 100%; height: auto; }`

3. **Media Queries:** Media queries apply different styles based on device characteristics like screen width, height, orientation, and resolution. This allows you to create specific styles for different devices.

Example: `@media (max-width: 600px) { .container { width: 100%; } }`

4. **Responsive Typography:** The text should be legible on all devices. Use relative units like ems or rems for font sizes and adjust line heights and spacing accordingly.

   Example: `body {font-size: 1em; line-height: 1.6; }`

5. **Touch-Friendly Design:** Ensure that buttons and links are large enough and spaced adequately to be easily tapped on touchscreens. Avoid hover effects that do not translate well to touch devices.

Introduction to Web Accessibility

Web accessibility refers to making websites and web applications usable by people with disabilities. This includes ensuring that content is accessible to individuals with visual, auditory, motor, and cognitive impairments. The goal is to create an inclusive digital environment where all users can interact with and benefit from web content regardless of their abilities.

Key Aspects of Web Accessibility

1. **Perceivable:** Information and user interface components must be presented to users in ways they can perceive. This includes providing text alternatives for non-text content, making content adaptable, and ensuring that content can be both seen and heard.
2. **Operable:** User interface components and navigation must be operable by all users. This means that all functionality must be accessible via a keyboard, and users must be able to interact with the content in various ways, for example, using the tab key on the keyboard to move around the different elements on the webpage.
3. **Understandable:** Information and the operation of the user interface must be understandable. Text should be readable and predictable, and the user interfaces should be consistent and simple.
4. **Robust:** Content must be robust enough to work with current and future technologies, including assistive technologies. This involves using valid HTML and following best practices to ensure compatibility.

Importance of Web Accessibility

1. **Inclusivity:** Accessibility ensures that people with disabilities have equal access to information and services, promoting inclusivity and equal opportunity.
2. **Legal Compliance:** Many countries have legal requirements for web accessibility, such as the Americans with Disabilities Act (ADA) in the U.S. or the Web Content Accessibility Guidelines (WCAG) internationally. Compliance helps avoid legal issues and ensures that your website meets legal standards.
3. **Enhanced User Experience:** Accessible design improves the overall user experience for everyone, not just those with disabilities. Features like clear navigation and readable text benefit all users.
4. **Broader Audience:** By making your website accessible, you reach a larger audience, including those using assistive technologies or experiencing temporary impairments.
5. **SEO Benefits:** Accessible websites often have better SEO because they follow best practices for structure and content, making it easier for search engines to index and rank them.

HINT

The lessons are to take learners through a practical exercise of identifying and assessing website accessibility. Learners can start with https//ges.gov.gh

Creating a Shopping Cart

Basic Structure of a Shopping Cart

A shopping cart is an essential component of any e-commerce website. It allows users to select products, view their selected items, and proceed to checkout. Here's a basic overview of the shopping cart structure:

1. **Cart Container:** A section or div where the cart items will be displayed.
2. **Item List:** A list showing the items added to the cart, including product details like name, quantity, and price.
3. **Cart Controls:** Buttons or controls for modifying item quantities, removing items, and proceeding to checkout.
4. **Summary:** A summary section displaying the total price and other relevant information.

LEARNING TASKS

1. Learners study the preparatory materials on e-commerce sites materials before their face-to-face class
2. Learners identify the features of the e-commerce site for their chosen company and subsequently apply the concepts in the tutorial to develop the site.
3. Learners must keep a journal reflecting on their design journey, challenges faced, and lessons learned throughout the project. They should highlight how they achieved the development goals and addressed responsiveness, accessibility, and user engagement.

PEDAGOGICAL EXEMPLARS

1. **Flipped Classroom:** Provide preparatory materials on e-commerce sites (video tutorials and interactive exercises) for learners to study at home. Use class time for hands-on practice/ project on developing an e-commerce website under the teacher's guidance.
2. **Interactive Lecture and Discussion:** Use the review and discussion of materials provided earlier, to examine popular e-commerce platforms and websites learners know (e.g., Amazon, Jumia). Use multimedia (images, videos) to illustrate the concept of e-commerce, its benefits, and key components, including product presentation, checkout processes, responsive design and accessibility.
3. **Hands-on Practical – Group Work:** Put learners into mixed-ability groups. Each group will choose a local company or create a fictional brand for which they will design an e-commerce web page and compete against other groups for the best-designed site.

   Provide a tutorial and sample HTML code on using a web editor to create web pages, focusing on key features like product catalogues and checkout processes.

KEY ASSESSMENT

Level 1 (Recall and Reproduction)

1. What are some key features of an e-commerce website
2. Identify some common elements found on a product page.

Level 2 (Skill, Concept and Applying Procedures)

1. Describe the steps in creating a basic product page for an e-commerce website.
2. Explain the purpose of a product search bar on an e-commerce website.

Level 3 (Strategic Thinking and Reasoning)

1. Discuss how product categorisation and filtering options can improve user experience in an e-commerce website.
2. Using virtual reality, imagine you are tasked with designing an e-commerce website for a company that sells sporting goods. How can you tailor the design and functionality to cater to the specific needs of their target audience.

Level 4 (Extended Thinking and Reasoning)

Imagine you are tasked with designing an e-commerce website for a company that sells sporting goods. How can you tailor the design and functionality to cater to the specific needs of their target audience? Thinking about accessibility and ease of use.

HINT

This week's recommended mode of assessment **is virtual reality,** *using Dok level 3 question 2 as a sample. Refer to the Teacher Assessment Manual and Toolkit page 70 for information on how to use this assessment mode.*

WEEK 22

Learning Indicator: Design an e-commerce web page for a selected company or organisation

FOCAL AREAS

1. Payment Gateway Integration
2. Shopping Cart and Checkout Process

Payment Gateway Integration

Payment Gateways are essential components of online payment systems. They securely transmit transaction information between a customer's and merchant's banks. Here is a breakdown:

A payment gateway is a service that authorises and processes online payments. It acts as a bridge between the customer's payment information and the merchant's bank account.

Importance of Payment

1. **Security:** Payment gateways encrypt sensitive payment data, ensuring secure transactions.
2. **Convenience**: They facilitate various payment methods, including credit/debit cards and digital wallets.
3. **Efficiency:** Payment gateways handle authorisation and processing quickly, allowing for immediate transaction confirmation.
4. **Compliance:** They ensure transactions comply with financial regulations and standards, such as PCI-DSS (Payment Card Industry Data Security Standard).

Examples of Popular Payment Gateways

1. **PayPal is o**ne of the most widely used payment gateways, offering easy integration and support for multiple payment methods. It is a secure way of conducting transactions, invoicing, recurring payments, and having global reach.
2. **Stripe is a** robust payment gateway known for its developer-friendly API (Application Programming Interface) and extensive features. It offers a customisable checkout experience, subscription billing, and support for various payment methods.
3. **Square:** Known for its simplicity and point-of-sale solutions. Square offers online payment processing. It involves an integrated POS, online payments, invoicing, and reporting tools.
4. **Authorize.Net** provides reliable payment processing with a wide range of features for businesses of all sizes. It is known for fraud detection, recurring billing and customer information management.

Discussion

1. Discuss common issues in form validation and how to handle them.
2. Explore additional validation options, such as server-side validation or more complex user feedback mechanisms.

LEARNING TASKS

1. Learners study the preparatory materials on e-commerce sites materials before their face-to-face class.
2. Learners actively engage in group discussions to identify the features of the e-commerce site for their chosen company and subsequently apply the concepts in the tutorial to develop the site.
3. Learners must continue with their journal, reflecting on their design journey, challenges faced, and lessons learned throughout the project. They should highlight how they achieved the development goals of implementing payment integration and shopping carts.

PEDAGOGICAL EXEMPLARS

1. **Flipped Classroom:** Provide preparatory materials on payment integration and shopping carts (video tutorials and interactive exercises) for learners to study at home. Use class time for hands-on practice/ project on developing an e-commerce website under the teacher's guidance.
2. **Hands-on practice (Group work) – Group Work**
   a. Guide learners to practice the process of integrating a mock payment gateway into their previously created e-commerce site, using sandbox environments provided by platforms like PayPal or Stripe. This hands-on experience will make the learning more relevant and applicable.
   b. Learners create a shopping cart on their web pages, including adding/removing items, updating quantities, and calculating totals (Use existing code and other online resources to help learners), emphasising security measures, such as HTTPS and SSL certificates, to protect customer data.
3. **E-portfolio: Web Design Portfolio Assessment:** The Teacher puts learners into mixed-ability groups and guides learners to compile their web design projects into portfolios showcasing their skills and creativity. Use the portfolios to assess learners' understanding and application of web design principles.

KEY ASSESSMENT

Level 1 (Recall and Reproduction)

1. What are the 4 popular payment gateways used on e-commerce sites?
2. Write down 1 benefit for each of the 4 gateways

Level 2 (Skill, Concept and Applying Procedures): Explain the purpose of a Payment Gateway on an e-commerce website

Level 3 (Strategic Thinking and Reasoning)

1. How can high-quality product images and detailed descriptions effectively increase customer trust and purchase likelihood on an e-commerce website?
2. Explain the importance of a clear and secure checkout process on an e-commerce website.

Level 4 (Extended Thinking and Reasoning)

1. Analyse the impact of mobile responsiveness on e-commerce websites. How can you ensure your design provides an optimal user experience for users browsing smartphones or tablets?
2. Discuss the importance of search engine optimisation (SEO) for e-commerce websites. How can you optimise product pages and website content to rank higher in search engine results and attract more organic traffic?

HINT

*This week's recommended mode of assessment is **questioning**, using Dok level 3 question 2 as a sample.*

WEEK 23

Learning Indicator: Design an e-commerce web page for a selected company or organisation

FOCAL AREAS

1. User Account Management
2. Security and Trustworthiness

1. **Enhancing User Experience:** Teachers should help learners understand how user accounts allow for personalised experiences, such as customised recommendations, order history and saved preferences. They also provide convenience by allowing users to store their shipping addresses, payment methods, and other details, making future purchases faster and easier.

2. **Order Tracking:** This allows customers to view past orders, track shipments, and manage returns through their accounts. Learners should also understand that e-commerce provides real-time updates on the status of their orders, improving transparency and customer satisfaction.
3. **Customer Support:** It allows users to manage support requests, view responses, and track the progress of their issues. Customer service representatives can access user profiles to better support and resolve issues more efficiently.
4. **Security:** User accounts offer a secure way to access personal information and transaction history. Implementing secure login mechanisms allows authentication and reduces the risk of unauthorised access and fraud.

Basic User Account Features

1. **Account Creation:** Registration Forms: Users fill out registration forms with details such as name, email address, and password. This includes email verification or CAPTCHA to ensure authenticity and reduce spam.

   CAPTCHA

   CAPTCHA, which stands for Completely Automated Public Turing Test to Tell Computers and Humans Apart, is a challenge-response test that acts as an authentication mechanism on websites, search engines, and web applications to ensure that users (with or without password-based credentials) are human beings and not automated bots trying to crowd the system and cause a cyberattack. Figure 22.1 below shows an example of CAPTCHA.

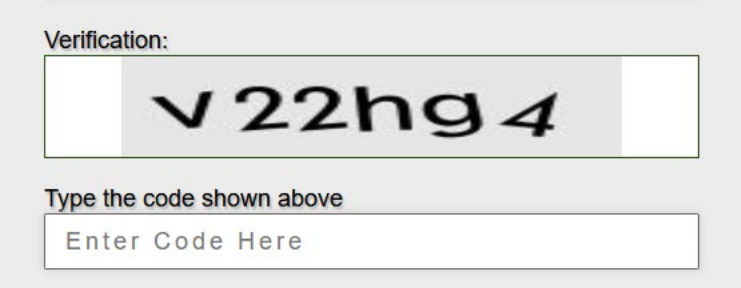


Figure 22.1: *CAPTCHA*

2. **Login:** Users enter their credentials (username/email and password) to access their accounts. The secure login process includes password hashing/salting and potentially multi-factor authentication (MFA).

Creating User Accounts

Structuring Account Creation Forms

1. **Overview:** Account creation forms are critical for allowing users to register for an account on a website. These forms collect necessary information such as username, email, and password.
2. **Form Structure**
   a. Form Elements: Typically includes text fields, email fields, password fields, and submit buttons.
   b. Labels and Inputs: Each form field should have a corresponding label for accessibility and clarity.
   c. Submit Button: Allows users to submit their information.

Figure 22.2 Structure of a form

Managing User Accounts

Viewing and Editing Profile Information

Profile Management

1. Teachers should guide learners to understand that user profiles allow individuals to view and update their personal information, such as names, email addresses, and passwords.
2. Key Features include viewing profile information by displaying current details like username, email, and address. It also allows users to modify and save changes to their details.

Security and Trustworthiness

Why Security Matters

Protection of Sensitive Information is an essential part of e-commerce websites. The website handles sensitive data such as customer personal details, payment information, and order histories. Ensuring this data is protected from unauthorised access is crucial.

Teachers should help learners understand that **Trust and Reputation matter a lot in e-commerce.** Security breaches can damage a company's reputation and erode customer trust. Customers are less likely to engage with a site if they feel their information is at risk. The company is also at risk of being fined if they do not include appropriate security principles and a serious data breach occurs.

Ghana and many other countries have regulations requiring secure personal information handling (e.g., the Data Protection Act). Non-compliance can lead to legal consequences and fines.

Learners should consider the following key security principles when designing an e-commerce website.

1. **Confidentiality: designers and admins must e**nsure that information is only accessible to those authorised to view it.
2. **Integrity:** Protecting information from being altered by unauthorised individuals.
3. **Availability:** Ensuring that information and services are available to authorised users when needed.

Secure Authentication Mechanisms

Password Hashing and Salting

Password Hashing

1. Hashing is a process that converts a password into a fixed-size string of characters, which appears random. It is a one-way function, meaning it cannot be reversed to obtain the original password.
2. Hashing passwords ensures that even if the password data is compromised, the actual passwords are not easily retrievable.

Password Salting

1. Salting involves adding a unique, random string (the salt) to a password before hashing. This prevents attacks where attackers guess passwords using precomputed hash tables (rainbow tables).
2. Salting ensures that even if two users have the same password, their hashed passwords will differ due to the unique salt.

Implementing Two-Factor Authentication

Two-factor authentication (2FA) adds an extra layer of security by requiring users to provide two forms of identification: something they know (password) and something they have (a temporary code sent to their device or email address).

How 2FA Works

Step 1: The user logs in with their username and password.

Step 2: After successful authentication, the user is prompted to enter a second factor, such as a code sent via SMS, email or generated by an authenticator app.

Step 3: The system verifies the second factor. If correct, the user is granted access.

LEARNING TASKS

1. Learners listen to and interact with teachers and peers, take notes, and ask questions to clarify their understanding of user account management and e-commerce security.
2. Learners develop the sample website they have been assigned.

PEDAGOGICAL EXEMPLARS

1. **Case Study Exploration and Discussion:** Using video analysis, the Teacher leads learners to analyse user account management features and security measures using popular e-commerce platforms (e.g., Amazon, Jumia) as a case. The teacher will discuss learners' experiences with user accounts on e-commerce sites.
2. **Hands-on practice (Group work):** The teacher divides learners into mixed-ability groups and guides them in applying web design principles, replicating user account management and the security features discussed for their e-commerce web page.

KEY ASSESSMENT

Level 1: Recall and Reproduction

1. What does SSL stand for in web security?
2. Explain the term user authentication.
3. What is encryption?
4. What does HTTPS stand for.
5. Name two common security features used in e-commerce websites to protect user data.

Level 2: Skills and Concepts

1. Explain why HTTPS is important for securing e-commerce transactions.
2. Explain the difference between a secure password policy and a password reset process in e-commerce user account management.
3. Describe how two-factor authentication enhances user account security on e-commerce platforms.
4. Describe why HTTPS is important for securing e-commerce transactions.

Level 3: Strategic Thinking

1. Analyse the potential risks and benefits of implementing biometric authentication for user accounts on an e-commerce website.
2. Evaluate the impact of using social media logins (e.g., Facebook or Google) for user account creation on e-commerce sites. Discuss the pros and cons.

Level 4: Extended Thinking

1. Develop a comprehensive security strategy for an e-commerce website, focusing on user account management, data protection, and fraud prevention. Include key components and justify your choices.
2. Research a recent high-profile security breach in the e-commerce industry and propose measures that could have prevented it. Analyse the breach's impact and the effectiveness of your proposed solutions.
3. You are presented with an existing ecommerce website. Identify potential improvements to user interface (UI), user experience (UX), and conversion optimisation. Explain your reasoning and suggest design changes to enhance user engagement and sales.

HINT

*This week's recommended mode of assessment is **discussion**, using Dok level 2 question 4 as a sample. Refer to the Teacher Assessment Manual and Toolkit page 52 for information on how to use this assessment mode.*

WEEK 24

Learning Indicator: Create a relational database model of a database

FOCAL AREA: DATABASE DESIGN

Database Concepts

A relational database organises data into structured tables related to each other through shared data values. This organisation is crucial for efficiently managing large volumes of data, ensuring data integrity, and facilitating complex queries and data analysis. For an e-commerce website, this means seamlessly handling product catalogues, customer information, orders, and transactions.

Importance of a Database

A well-designed relational database is important for an e-commerce website because it:

1. Efficiently Manages Large Volumes of Data - Organises and stores extensive data collections in an orderly manner.
2. Ensures Data Integrity and Reduces Redundancy - Maintains accuracy and consistency, preventing duplicate data entries.
3. Facilitates Complex Queries and Data Analysis - Supports advanced data retrieval and analysis for better decision-making.
4. Dynamic and Responsive Functioning—This function handles multiple aspects of e-commerce, such as product listings, customer details, orders, and transactions, making the website dynamic and responsive.

Key Components of a Relational Database

Tables (Relations)

A table is a collection of related data entries consisting of columns and rows. Each table in a database holds data about a specific subject. For example:

Product Table - Contains data about items for sale.

Customer Table - Stores information about individuals who purchase products.

Order Table - Holds records of transactions between customers and the store.

Transaction Table - Contains details of payments made by customers.

Columns (Fields/Attributes)

Columns define a table's structure, with each column representing a data attribute. For instance, in a "Customers" table, the columns might include CustomerID, Name, and Email.

Customer ID	Name	Email
1	Kwasi Anokye	anokye@example.com
2	Senyo Dordoe	dordoe@example.com

Rows (Records/Tuples)

Rows are individual entries in a table, each representing a single record. Each row contains a unique data instance for the attributes the columns define. Below is an observation from our customer table.

1 | Kwasi Anokye | anokye@example.com

Entities and Attributes

Entities are objects or concepts about which data is stored in a database, and attributes are the properties or details of an entity. For an e-commerce website:

Entities

- **Product** - Items for sale.
- **Customer** - Individuals who purchase products.
- **Order** - Transactions between customers and the store.
- **Transaction** - Details of payments made by customers.

Attributes

Attributes are the columns in a database table that define the properties of the entities stored in the table. Each attribute holds specific data values that pertain to a particular aspect of the entity.

Product - ProductID, Name, Description, Price, StockQuantity.

Customer - CustomerID, Name, Email, Address, PhoneNumber.

Order - OrderID, CustomerID, OrderDate, OrderTotal.

Transaction - TransactionID, OrderID, PaymentMethod, TransactionDate, Amount.

Relationships between Entities

Relationships link tables based on shared data. In a relational database, these relationships can be:

- One-to-Many - A single customer can place many orders.
- Many-to-Many - Products can appear in multiple orders, and orders can contain multiple products.

Primary Key

A primary key is a field (or a combination of fields) in a database table that uniquely identifies each row or record within that table. It ensures that each record can be uniquely distinguished from all others.

Key Characteristics

1. **Uniqueness** - Every primary key value must be unique within the table. This uniqueness ensures that each record can be individually identified.
2. **Non-null** - A primary key cannot contain NULL (empty) values. This requirement ensures that every record has a valid and identifiable primary key value.
3. **Immutability** - Ideally, the value of a primary key should not change. Once assigned, it should remain the same for the lifetime of the record.

Importance of Primary Keys

1. **Ensuring Data Integrity**—The primary key enforces uniqueness, preventing duplicate records from existing within the table.
2. **Efficient Data Retrieval** - Database systems use primary keys to locate and retrieve data quickly. This efficiency improves query performance.
3. **Establishing Relationships** - Primary keys are often referenced by foreign keys in other tables to create relationships between tables. This referencing helps maintain referential integrity.

Foreign Key

A foreign key is a column or set of columns in one table that refers to the primary key in another table. It creates a relationship between the two tables, allowing data from one table to be associated with data from another.

Key Characteristics

1. **Referential Integrity** - Foreign keys help maintain referential integrity by ensuring that the value in the foreign key column corresponds to a valid value in the referenced table's primary key column.
2. **Nullable** - Foreign key columns can contain NULL (empty) values, meaning a record might not always have a related record in the referenced table.
3. **Consistency** - The foreign key enforces consistency between the two related tables, ensuring the data remains accurate and reliable.

Importance of Foreign Keys

1. **Establishing Relationships** - Foreign keys define relationships between tables, such as one-to-many and many-to-many relationships.
2. **Maintaining Data Integrity** -They ensure that relationships between tables are consistent and valid, preventing orphaned records or invalid references.
3. **Facilitating Joins** - Foreign keys join tables in queries, allowing for complex data retrieval and analysis.

Examples

Consider an e-commerce database with the following tables

1. **Customer Table**

   Primary Key – CustomerID

CustomerID	Name	Email
1	Kwasi Anokye	anokye@example.com
2	Senyo Dordoe	dordoe@example.com

2. **Order Table**

   Primary Key - OrderID

   Foreign Key - CustomerID (references the CustomerID in the Customer table)

OrderID	OrderDate	CustomerID
101	2024-07-01	1
102	2024-07-02	1
103	2024-07-01	2

In the Order table, CustomerID is a foreign key that references the CustomerID in the Customer table. This relationship ensures that each order is associated with a valid customer.

OrderItems Table

Primary Key - Composite key of OrderID and ProductID

Foreign Keys

OrderID (references OrderID in the Order table)

ProductID (references ProductID in the Product table)

OrderID	ProductID	Quantity
103	A1	2
102	B2	1
102	A1	1

In the OrderItems table, OrderID is a foreign key that references OrderID in the Order table, linking each item to a specific order.

ProductID is a foreign key that references ProductID in the Product table, indicating which products are included in each order.

Relational Database Tables and Fields

Tables and fields form the backbone of a relational database. Each table is a collection of related data entries, and each field is a column that holds specific information about every record in the table. For instance:

Product Table - Fields might include ProductID, Name, Description, Price, and StockQuantity.

Customer Table - Fields might include CustomerID, Name, Email, Address, and PhoneNumber.

Order Table - Fields might include OrderID, CustomerID, OrderDate, OrderTotal.

Transaction Table - Fields might include TransactionID, OrderID, PaymentMethod, TransactionDate, and Amount.

Schema Design

Schema Design is a conceptual blueprint that outlines how data is organised within a database. It involves defining tables, fields, relationships, and constraints to ensure data integrity and efficient access.

ProductID

Type - INT

Purpose - Unique identifier for the product.

Constraints - PRIMARY KEY ensures uniqueness.

Name

Type - VARCHAR(255)

Purpose - Product name.

Constraints - NOT NULL ensures that every product has a name.

Description

Type - TEXT

Purpose - Detailed description of the product.

Constraints - This field can be optional (NULL allowed).

Price

Type - DECIMAL(10, 2)

Purpose - Product price.

Constraints - NOT NULL, CHECK (Price >= 0) ensures non-negative pricing.

StockQuantity

Type - INT

Purpose - Number of units available.

Constraints - NOT NULL, CHECK (StockQuantity >= 0) ensures non-negative stock levels.

CategoryID

Type - INT

Purpose - Links to a Category table if categories are used.

Constraints - FOREIGN KEY if there is a Category table.

Brand

Type - VARCHAR(100)

Purpose - Product brand name.

Constraints - Optional field.

Query

A request for data or information from a database. SQL queries are written to perform various operations such as retrieving, updating, or deleting data.

Example: SELECT * FROM Product WHERE Price > 500;

This query will select all fields (* means ALL) from the product table, for all records where the price is greater than 500

SELECT Statement

A SQL command is used to retrieve data from one or more tables. It specifies the columns to be retrieved and the conditions for selecting records.

Example: SELECT Name, Price FROM Product WHERE StockQuantity > 10;

This query will select the name and the price from the product table, for all records where the stock quality is greater than 10

INSERT Statement

An SQL command is used to add new records to a table. It specifies the table and the values for each column.

Example: INSERT INTO Product (ProductID, Name, Price, StockQuantity) VALUES (1, 'Laptop', 999.99, 50);

This query will insert a new record into the product table with the given values

UPDATE Statement

A SQL command is used to modify existing records in a table. It specifies the table, columns to be updated, and the conditions for updating records.

Example: UPDATE Product SET Price = 899.99 WHERE ProductID = 1;

This query will change the price of product with an ID of 1 within the product table to be 899.99

DELETE Statement

An SQL command is used to remove records from a table. It specifies the table and the conditions for deleting records.

e.g DELETE FROM Product WHERE ProductID = 1;

This query will delete product with an ID of 1 from the product table

WHERE Clause

An SQL clause specifies conditions that filter records in a query. It is used with SELECT, UPDATE, and DELETE statements.

e.g SELECT * FROM Product WHERE Price < 500;

This query will select all fields from the product table, for all products where the price is less than 500

ORDER BY Clause

A SQL clause is used to sort a query's results based on one or more columns. You can specify ascending (ASC) or descending (DESC) order.

e.g SELECT * FROM Product ORDER BY Price DESC;

This query will select all fields from the product table and order them in descending order based on the price

GROUP BY Clause

A SQL clause used to group rows that have the same values in specified columns into summary rows. It is often used with aggregate functions.

e.g SELECT CategoryID, COUNT(*) AS NumberOfProducts FROM Product GROUP BY CategoryID;

JOIN

A SQL operation combines rows from two or more tables based on a related column. Types of joins include INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN.

e.g SELECT Product.Name, Category.CategoryName FROM Product INNER JOIN Category ON Product.CategoryID = Category.CategoryID

Aggregate Functions

This function performs calculations on a set of values and returns a single value. Common aggregate functions include COUNT(), SUM(), AVG(), MAX(), and MIN().

e.g SELECT AVG(Price) AS AveragePrice FROM Product;

Constraint

Rules applied to table columns to ensure the validity and integrity of data. Constraints include NOT NULL, UNIQUE, CHECK, and FOREIGN KEY.

e.g CHECK (Price >= 0) ensures that the price cannot be negative.

Scan for examples

LEARNING TASKS

1. Learners listen to and interact with teachers and peers, take notes, and ask questions to clarify their basic database concepts, designing and building a database for an e-commerce website.
2. Learners design and develop a database for their e-commerce

PEDAGOGICAL EXEMPLARS

1. **Flipped Classroom:** Provide preparatory materials (video tutorials and interactive exercises) on basic concepts and techniques to build for learners to study at home. Use class time for hands-on practice/project to answer questions on basic concepts and design and build a database for an e-commerce website under the teacher's guidance.
2. **Hands-on practice (Group work) – Group Work:** The Teacher puts learners in mixed-ability groups and guides learners through the practical process of building and integrating a database with an e-commerce website using a database management system (DBMS) like MySQL.

KEY ASSESSMENT

Level 1: Recall and Reproduction

1. What is a primary key in a relational database?
2. Name two common relational database management systems (RDBMS).

Level 2: Skills and Concepts

1. Explain the role of foreign keys in maintaining relationships between tables in a relational database.
2. Describe how you would create a simple relational database model for an e-commerce site, including tables for customers, orders, and products.

Level 3: Strategic Thinking

1. Analyse the potential challenges of integrating a relational database with an e-commerce website and propose solutions to address these challenges.
2. Evaluate the benefits and drawbacks of using a relational database versus a NoSQL database for an e-commerce platform.

Level 4: Extended Thinking

1. Design a comprehensive relational database model for an e-commerce platform, detailing the required tables, fields, and relationships. Explain how this model supports various e-commerce functions, such as inventory management, customer service, and order processing.
2. Conduct a case study on a successful e-commerce platform's database architecture. Analyse the key features of its database design, how it handles large volumes of transactions, and the measures taken to ensure data security and integrity.

HINT

The Recommended Mode of Assessment for Week 24 is End of Semester Examination. Refer to ***Appendix G*** *for a Table of Specification to guide you to set the questions. Set questions to cover all the indicators covered for at least Weeks 13 to 24.*

SECTION 6 REVIEW

In this section, learners explore the fundamentals of web page editors, focusing on design elements, sections, text, images, GUI/UI/UX and multimedia integration. They will understand basic ideas like web page editors, HTML and Cascading Style Sheets (CSS), page structure and layout, formatting and typography, etc. They acquire skills to get the most out of programming logic and solve complex computational logic problems. By mastering these ideas, learners will be ready for further studies and career opportunities in computer science. The content connects with mathematics and electrical engineering. Weekly topics include web page editors and design elements (Week 19), GUI/UI/UX principles, image and multimedia integration (Week 20), e-commerce web page design (Week 21), Payment Gateway Integration and Shopping Cart and Checkout Process (Week 22), User Account Management and Security trustworthiness (Week 23), and creating a relational database (Week 24). Diverse teaching pedagogies and Webb's Depth of Knowledge (DOK) assessment levels ensure comprehensive understanding and critical thinking.

APPENDIX G: END OF SEMESTER EXAMINATION

1. **Structure**

This end of semester examination covers focal areas from weeks 13-23 taking into consideration DOK levels. The test should include

a) Section A - Multiple Choice (40 questions answer all)

b) Section B - (5 Essay questions, 3 to be answered)

c) Test of Practical Knowledge (TPK) - (5 questions, answer 3)

2. **Resources**

a) Computer

b) Answer booklets

c) Learner Material

d) Teacher Manual

e) Teacher Assessment Manual and Toolkits

3. **Sample Question**

a. **Multiple Choice Question**

i. **What is the main purpose of a hash table in computer science?**

A. To store data in a sequential manner

B. To map keys to specific values quickly

C. To organise data alphabetically

D. To perform mathematical calculations

b. **Essay Type**

Discuss how product categorisation and filtering options can improve user experience in an e-commerce website

c. **Test of Practical**

Develop a custom webpage editor tool with advanced features beyond basic HTML and CSS editing

Marking Scheme and Rubrics

Section A - Multiple choice

Answer is B

1 mark for each correct answer

20 marks

Section B – Essay

Criteria	Need Improvement (1 mark)	Good (2 marks)	Excellent (3 marks)
Explanation of Improved User Experience	Gives a simple statement on user experience, e.g., 'Makes shopping easier.'	Mentions how categorisation or filtering aids navigation, e.g., 'Categories and filters help users find items faster.'	Explains how categorisation and filtering reduce frustration, e.g., 'By organising products and allowing filters, users can easily navigate and find what they need without scrolling through irrelevant products.'
Use of Examples	No examples provided, e.g., does not give specific categories or filters.	Provides a basic example, e.g., 'Filters like "price" and "size" help users choose.'	Gives relevant examples, e.g., 'A store selling electronics might categorise products into "Mobile Phones," "Laptops," and "Tablets," with filters for "price," "brand," and "features."'
Clarity and Structure of Response	Response is unclear and poorly organised, e.g., 'It helps users.'	Response is somewhat clear but lacks structure, e.g., 'Categories and filters make it easier to find stuff, but it's not always easy.'	Response is mostly clear and well-structured, e.g., 'Categories organise products, and filters allow for refined searches, reducing time spent looking for specific items.'

Total = 16 marks

Section C - Practical

S/N	CRITERIA	DETAILS	MARKS ALLOCATION
1.	Functionality (Understanding of the task/problem)	The editor includes multiple advanced features such as live preview, syntax highlighting, a built-in file manager, custom templates, and integration with external libraries or frameworks The editor allows for basic HTML and CSS editing and includes at least one additional feature such as live preview, syntax highlighting, or a code validator	2 1
2.	User Interface (Critical thinking)	The editor has an intuitive and user-friendly interface with advanced controls such as drag-and-drop functionality, customizable toolbars, and context-sensitive help The editor has a simple user interface that allows users to edit HTML and CSS with basic controls.	2 1

3.	*Code Quality* *(Creativity and Innovation)*	*The code is well-organised, efficient, and includes thorough comments and documentation. It follows best practices for readability and maintainability.* *The code is functional but may contain some inefficiencies or lack comments*	*2* *1*
4.	*Testing and Debugging* *(Analytical Skills)*	*The editor has been thoroughly tested for various use cases and edge cases, with no significant bugs. It includes debugging tools or error handling features.* *The editor has been tested for basic functionality with some minor bugs or issues.*	*2* *1*
5.	*Presentation* *(Creativity, Innovation and organisational skills)*	*The editor is presented with comprehensive documentation, including user guides, feature explanations, and installation instructions.* *The editor is presented with basic documentation or instructions.*	*2* *1*

4. **How to Administer (MCQ, ESSAY TYPE AND PRACTICAL)**

   a. Administer constructed MCQs in a controlled environment to prevent cheating.
   b. Provide clear directions/instructions to learners
   c. Emphasise the importance of planning, drafting, revising and finalising the work.
   d. Provide physical prompts or visual prompts to the learner who may require assistance
   e. Provide support and accommodations for learners with special needs, such as extra time or a quiet testing environment.
   f. Encourage learners to manage time efficiently to complete tasks within allocated timeframes.
   g. Encourage teamwork and effective communication if tasks involve group work.

5. **Feedback**

   (MCQ, ESSAY TYPE AND TPK)

   a. Score the essay consistently based on the established criteria to reduce subjectivity.
   b. Give oral and written comments to the learner in the provision of feedback.
   c. Encourage peer critiquing in the feedback process.
   d. Give oral and written comments to the learner in the provision of feedback.
   e. Encourage peer critiquing in the feedback process.
   f. Give targeted feedback individually, in small groups and whole class
   g. Encourage learners to reflect on their performance, review their work, and identify areas of improvement.

6. TABLE OF SPECIFICATION

Week	Focal Area	Type of Question	DoK Level				Total
			1	2	3	4	
13	Hash-based Structures, Hash Tables and Maps, Real-life applications of data structures	Multiple Choice	2	2	-	-	4
		Essay	-	-	1	-	1
14	Basic Programming Concepts	Multiple Choice	1	1	1	-	3
		Practical	-	1	1	-	2
15	Control Structures	Multiple Choice	1	1	1	-	3
16	Creating a Simple Calculator	Multiple Choice	1	2	-	-	3
		Practical	-	1	1		2
17	Understanding Machine Learning Concepts	Multiple Choice	2	1	1	-	4
18	Implementing machine learning and Ethical Consideration	Multiple Choice	1	2			3
19	Introduction to Web Development, Basic HTML and CSS, Page Structure and Layout, Text Formatting and Typography	Multiple Choice	1	2	1		4
		Practical		-	1		1
20	Image Integration, GUI/UI/UX Principles, Multimedia Integration	Multiple Choice	1	2	1		4
		Essay		-	1		1
21	Responsive Design & Accessibility Considerations, Product Presentation and Catalogue, Shopping Cart and Checkout Process	Multiple Choice	2	2	-		4
		Essay		-	1		1
22	Gateway Integration Payment, Shopping Cart and Checkout Process	Multiple Choice	1	2	1		4
		Essay	-	-	1		1
23	User Account Management, Security and Trustworthiness	Multiple Choice	2	1	1		4
		Essay	-	-	1		1
	Total		15	20	15	-	50

REFERENCES

- Gregg, J. (1998). Ones and zeros: Understanding Boolean algebra, digital circuits, and the logic of sets. IEEE Press**.**
- Gregg, J. (1998). Ones and zeros: Understanding Boolean algebra, digital circuits, and the logic of sets. IEEE Press.
- Barcodes Ghana. (2020). Guide to retail barcodes Ghana. Barcodes Ghana. Retrieved June 20, 2024, from https://barcodesghana.com/wp-content/uploads/sites/96/2020/10/Guide-to-retail-barcodes-Ghana.pdf
- Qodenext. (no date). What is UPC Barcode - Explained in detail. Qodenext. Retrieved June 20, 2024, from https://www.qodenext.com/blog/what-is-upc/
- Redicode. (n.d.). EAN-13 Barcode Images. Redicode. Retrieved June 20, 2024, from https://redicode.com.au/product/ean-13-barcode-image/
- https://www.spiceworks.com/it-security/network-security/articles/what-is-captcha/