MINISTRY OF EDUCATION

GHANA ASSOCIATION OF SCIENCE TEACHERS

(GAST)

Computing

for Senior High Schools

Year 2

Raphael Dordoe Senyo

Anibrika Mark

Daniel Miheso

MINISTRY OF EDUCATION

GHANA ASSOCIATION OF SCIENCE TEACHERS

Computing

for Senior High Schools

Year 2

Raphael Dordoe Senyo

Anibrika Mark

Daniel Miheso

Ghana Education Service (GES)

ISBN: 978–9988–9493–1–0

CONTENTS

FOREWORD

Ghana's new Senior High School Curriculum aims to ensure that all learners achieve their potential by equipping them with 21st Century skills, knowledge, character qualities and shared Ghanaian values. This will prepare learners to live a responsible adult life, progress to further studies and enter the world of work. This is the first time that Ghana has developed a Senior High School 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.

The Ministry of Education is proud to have overseen the production of these Learner Materials which can be used in class and for self-study and revision. These materials have been developed through a partnership between the Ghana Education Service, teacher unions (Ghana National Association of Teachers-GNAT, National Association of Graduate Teacher -NAGRAT and the Coalition of Concerned Teachers- CCT) and National Subject Associations. These materials are informative and of high quality because they have been written by teachers for teachers with the expert backing of each subject association.

I believe that, if used appropriately, these materials 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.

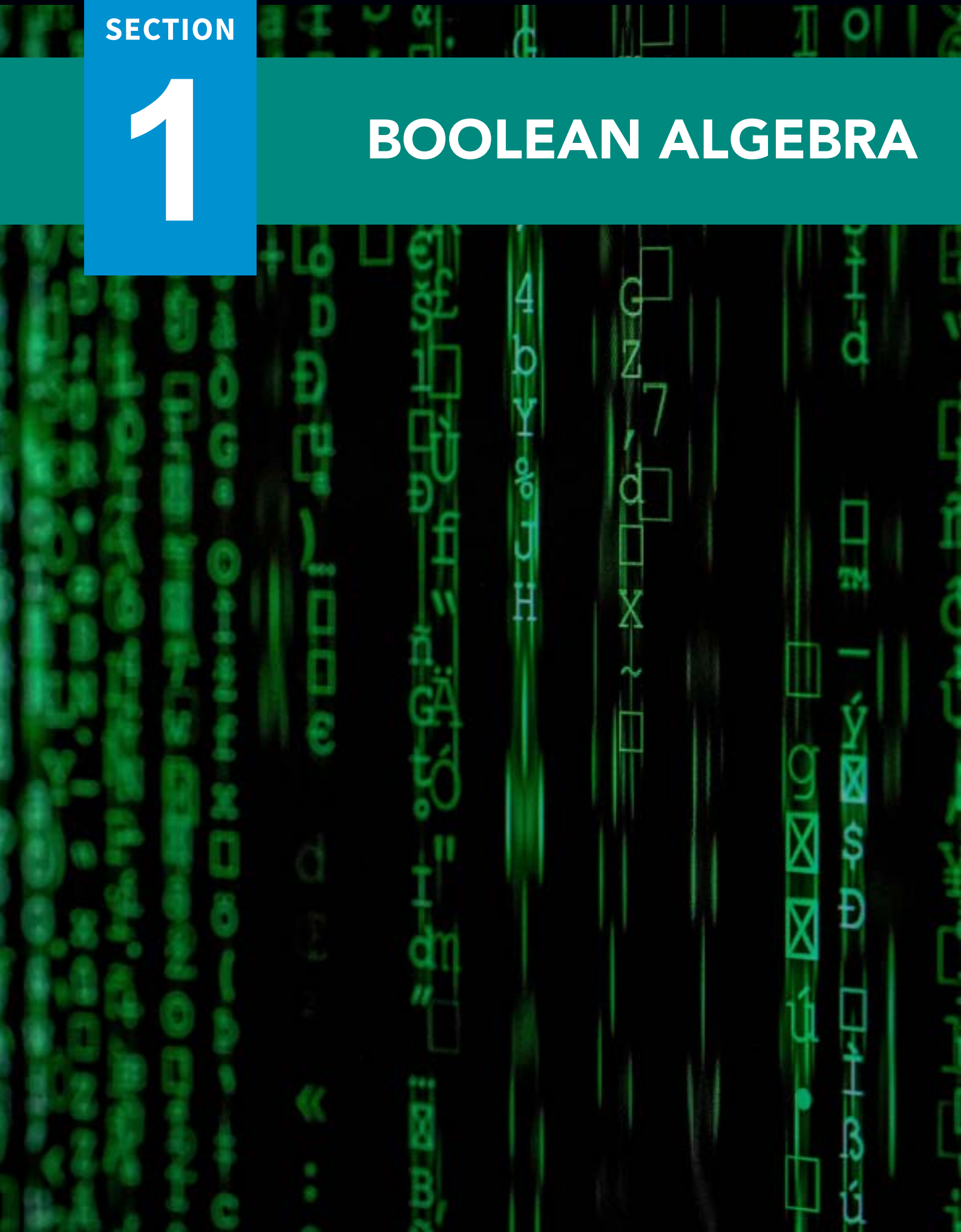
Haruna Iddrisu MP

Minister for Education

SECTION

1

BOOLEAN ALGEBRA



COMPUTER ARCHITECTURE & ORGANISATION

Data Storage and Manipulation

INTRODUCTION

Welcome to the second year of the computing course. In this section, we will learn about **Boolean logic** and **logic gates**, which help computers make decisions just like you do when you play games! Imagine tiny switches inside a computer that can say "yes" or "no" to help it work faster. These switches are called **logic gates,** connected through **circuits** like a network.

Stay curious to also learn about **Boolean expressions**, which are like secret codes and learn how computers quickly solve **arithmetic operations** like adding and multiplying.

Ready, steady, let's go learn and do some amazing, cool activities. Remember to ask questions when you do not understand a concept.

Key Ideas

- **Boolean logic** is a way of making decisions using simple statements that are either true or false.
- **Logic gates** are tiny parts inside a computer that make decisions based on the input they receive.
- **Circuits** are paths that allow signals to flow through and make things work in computers.
- **A Boolean expression** is a way of writing a rule or decision using true or false statements.
- **Arithmetic operations** are the addition, subtraction, multiplication and division used in computing mathematical tasks.

UNDERSTANDING LOGIC OPERATIONS AND EXPRESSIONS

Let us recap our learning on logic operations by doing this activity.

Activity 1.1

Recap of Logic Operation

1. Complete the table below with the correct outputs for AND, OR and NOT operations in groups.

A (Input)	B (Input)	AND (Output)	OR (Output)	NOT A (output)	NOT B (output)
0	0				
0	1				
1	0				
1	1				

2. Present your results to the class for discussion.

NAND Gate (Not AND)

The output of the activity you have just finished is called a NAND gate. A NAND gate is the result of combining an AND gate and a NOT gate. Its use is to reverse an AND gate. For instance, in Mr. Mark's house, when the water tank is empty or full, a sensor sends a signal using a NAND gate to control the pump to switch it on or off automatically.

The typical truth table of 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

Activity 1.2

Understanding NAND Gate

1. Copy and fill the first column in the table below with the correct output from **Activity 1.1**.

AND (Input)	~ AND (Output)

2. Apply the NOT operator to the first column and write the result in the second column.
3. Show your results to the teacher for approval.

NOR Gate (Not OR)

Similarly to the NAND gate, when you apply the NOT logic to the OR logic, the result is an NOR gate, just like you will do in **Activity 1.3**.

Have you ever wondered why some streetlights turn on and off automatically? It is because they use the NOR gates in circuits to control the lighting system, turning them on or off whenever the environment is dark or bright.

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

Activity 1.3

Understanding NOR Gate

1. Copy and fill the first column in the table below with the correct output from **Activity 1.1**.

OR (Input)	~ OR (Output)

2. Apply the NOT operator to the first column and write the result in the second column.
3. Show your results to the teacher for approval.

XOR Gate (eXclusive OR)

The combination of (~A AND B) and (A AND ~B) with the OR logic operator is called the "exclusive OR" or the "XOR" gate.

Note that **Activity 1.4** is the truth table for the XOR gate. An example of the XOR gate in real life is that some banks in Ghana use passcodes at Automated Teller Machines (ATMs). The XOR gate compares the input passcode with the stored passcode and if they are different, the ATMs locks the card; if they match, the lock opens.

Activity 1.4

Understanding XOR Gate

1. Complete the table below by using the NOT and OR logic at the approach column

A	~ A	B	~ B	(~A AND B) = M	(A AND ~B) = N	M OR N
0		0				
0		1				
1		0				
1		1				

2. Show your results to the teacher for approval.

Standard Symbols of Logic Gates

Every logic gate has both inputs and outputs. On the right side of the gate are the inputs and on the left side are the outputs, as shown in **Figure 1.3**.

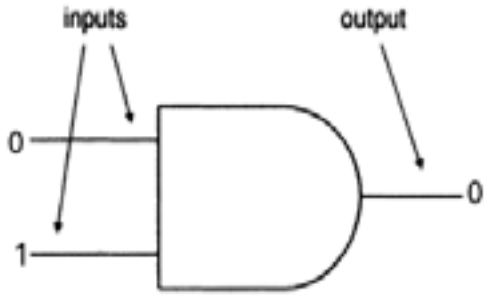


Figure 1.3: Input and output labels of a logic gate

So far, we have learnt six different logic gates, namely, AND, OR, NOT, NAND, NOR and XOR. We will now learn about their respective symbols, as shown in **Figure 1.4**.

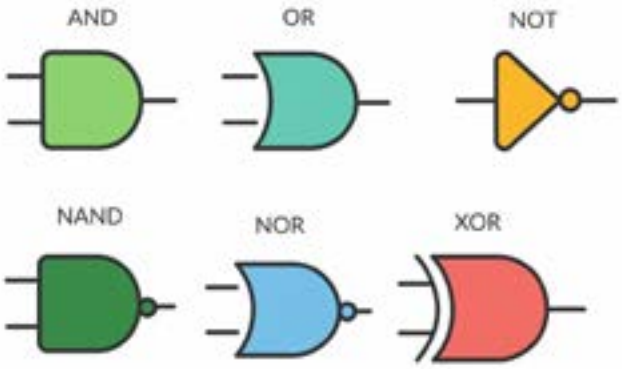


Figure 1.4: Symbols of Logic Gate

Activity 1.5

Drawing of Logic Gate

1. Observe **Figure 1.4** and draw the logic gate symbol for the following
   a. AND
   b. OR
   c. NOT
   d. NAND
   e. NOR

f. XOR

2. What similarities and differences have you noticed to enable you to easily differentiate the logic gates?
3. Match the logic gate name to the correct symbol from the table below

Gate Name	XOR	NOT	AND	OR	NAND	AND
Logic Symbol						

4. Share your results with the class for justification

Combination of Gates

You have done so well if you have been able to complete **Activities 1.4** and **1.5**. We are now going to combine two or more of the logic gates we have learnt to produce some interesting binary.

First, let us observe the behaviour of the NAND logic gate in the diagram below.

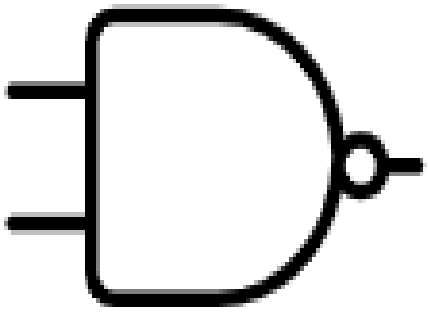


Figure 1.5: OR Logic gate

You may have noticed that the inputs 0 and 1 have the output 1, which is the same as the output shown in the truth table in **Figure 1.1**. It is important to know that the binary input and output of the logic gates are the same as their truth table.

Let us now observe how to combine the AND and NOT gates and see what it produces with 1 and 0 as the inputs for the AND gate, the result of 1 is shown in **Figure 1.6** below.

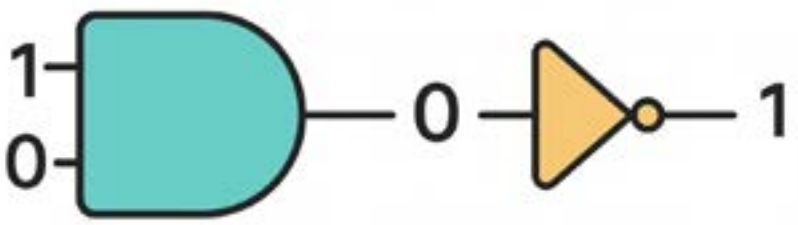


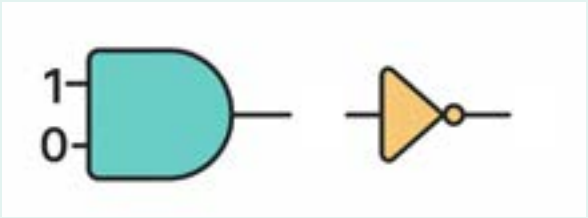
Figure 1.6: Combined Gate

Activity 1.6

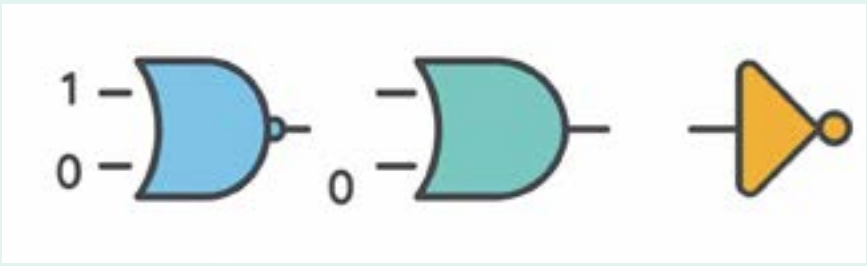
Combination of Gates

Work with your peers to complete the following task by writing the input and output binary for the following combinations:

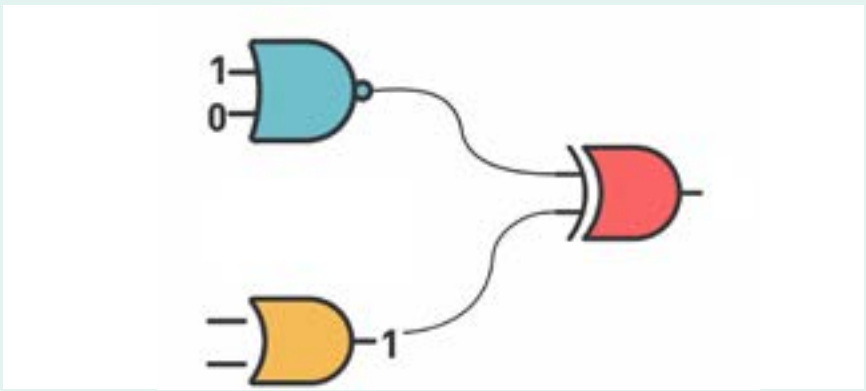
1.



2.



3.



4. Show your design to your teacher for approval.

Intermediate Stages

You may have noticed that before getting the final results, you have to pass through a number of gates; these gate(s) are called "intermediate stages." They make it easier to understand how logic moves through the gates for effective design, analysis and troubleshooting.

Activity 1.7

Problem Solving

1. Work with your peers to design a logic circuit that takes three inputs, A, B and C, and outputs a signal according to the following conditions:
   a. The output should be HIGH (1) only if:
      i. A and B are both HIGH (1), OR
      ii. A is LOW (0) and C is HIGH (1).
2. Show your design to your teacher for approval.

Hint: Use any of the following set of gates in the brackets: (AND, XOR and NAND) or (AND, NOT and XOR) to achieve the result.

BOOLEAN EXPRESSION

A Boolean expression is a mathematical statement that uses logical operators to evaluate binary values 0 (false) and 1 (true). These expressions are the language of digital logic, helping us describe how circuits behave based on different input combinations.

Boolean expressions use three main operators as shown in the table below.

Example of the use of boolean notations are:

$\wedge$ => A $\wedge$ B = A . B

$\vee$ => A $\vee$ B = A + B

There is another process that can solve the problem faster and easier; this is called "Boolean Expression" which you may have learnt in your first-year mathematics course.

Table 1.1: Shows the notation of the logic operations

Logic Operation	Boolean Notation
AND	$\wedge$
OR	$\vee$

Logic Operation	Boolean Notation
NOT	$\neg$ or ~

Identity Law

This law states that a variable would remain unchanged when it is ANDed with '1' or ORed with '0'. This is what was used in **Activity 1.8**.

Mathematically, we express the circuits as

(a) $A \wedge 1 = A \cdot 1 = A$

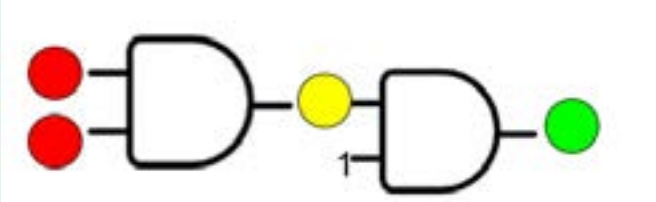
(b) $B \vee 0 = B + 0 = B$

Activity 1.8

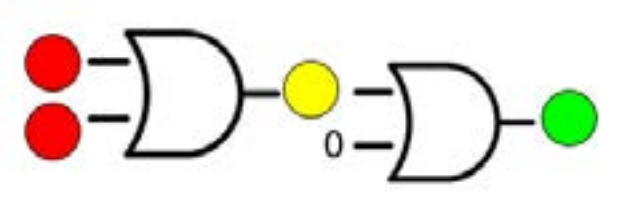
Identity Law

1. Fill the red circles with any binary inputs

   a.



   b.



2. Your output (yellow circle) from the intermediate gate is the same as the final answer (green circle).
3. Why do you think this is the case? Share your thought with the class.

De Morgan's Law

De Morgan's Laws show how negation (NOT) interacts with AND (·) and OR (+) in Boolean logic.

They allow you to transform one form of logical expression into another equivalent form."

An example is:

$(A \cdot B)' = A' + B'$

Where the AND (.) changes to OR (+). There are two main parts to this law:

1. First Law: If you say "NOT (A OR B)", like **Activity 1.8** (1) it means that both A and B are false. Rewritten as "NOT A AND NOT B".
2. Second Law: If you say "NOT (A AND B)," like **Activity 1.8** (2) it means at least one of them is false. This can be rewritten as "NOT A OR NOT B."

Activity 1.9

Inter-subject Learning

Work with your peers by applying your knowledge of De Morgan's law to solve the following:

1. $(A + B)' =$
2. $(A.B)' =$

Disjunctive Normal Form (DNF)

DNF is a way to write a Boolean expression so that it is easy to understand. It makes use of disjunction (OR); this means you are combining things with "OR." If any part is true, the whole thing is true.

Example:

1. (A **AND** B) **OR** (NOT A **AND** B) OR (A **AND** NOT B)
2. $(A \wedge B) \vee (\sim A \wedge B) \vee (A \wedge \sim B)$

Conjunction Normal Form (CNF)

CNF is a way to write a Boolean expression so that it is easy to understand. It makes use of conjunction (AND): This means you are combining things with "AND." All parts must be true for the whole thing to be true.

Example:

1. (A **OR** B) **AND** (NOT A **OR** B) **AND** (A **OR** NOT B)
2. $(A \vee B) \wedge (\sim A \vee B) \wedge (A \vee \sim B)$

Activity 1.10

Checking your understanding

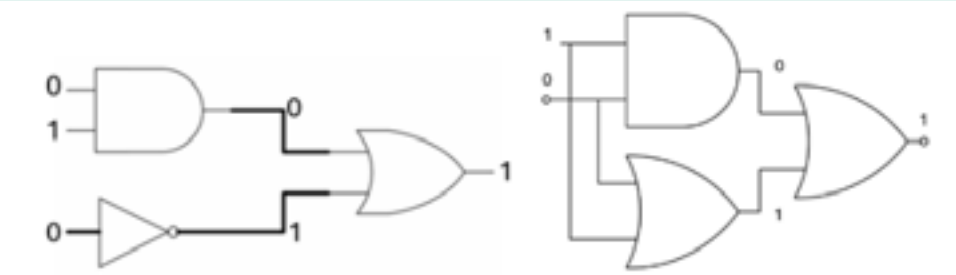
In less than 100 words, write in your own words the key difference between DNF and CNF.

Activity 1.11

Simplification of Circuits with Boolean Expression

Work with your peers to complete tasks 2, 3 and 4. Use task 1 as an example.

1.

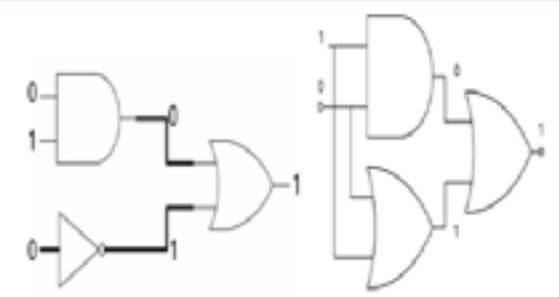


Gate1: $0 \wedge 1 = 0$

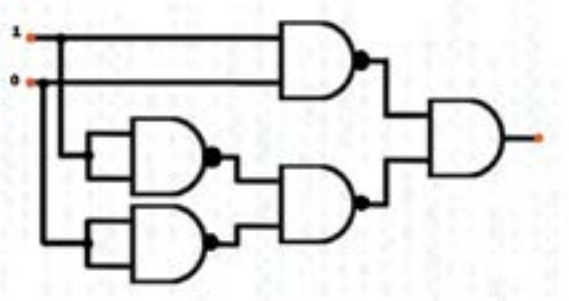
Gate2: $\sim 0 = 1$

Gate3: gate1result V gate2result $= 0$ V $1 = 1$

2.



3.

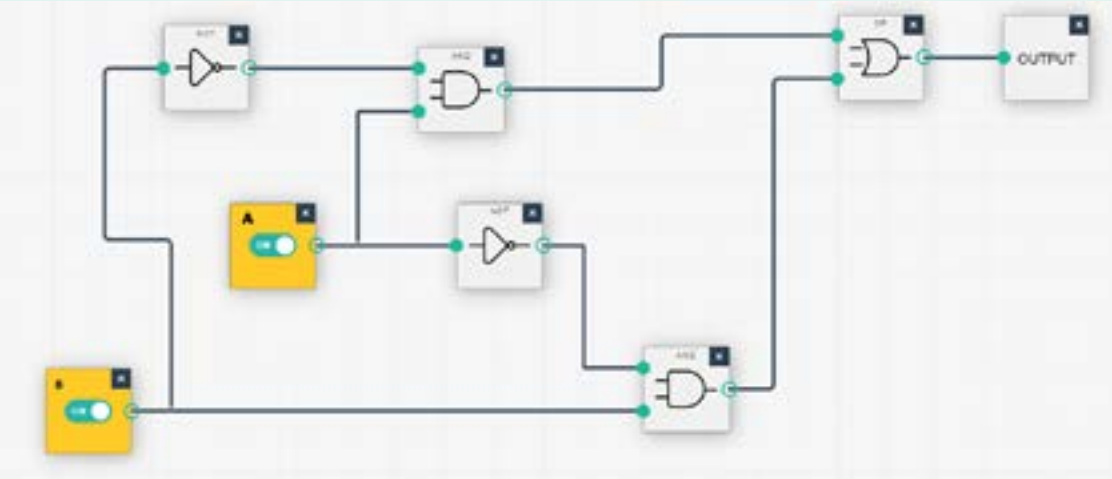


4. Show your results to the teacher

Activity 1.12

Simplification of Circuits Using DNF and CNF

1. Work with your peers to simplify the circuit below, using Boolean expression



2. Draw a circuit for Y = B V A
3. Write less than 10 words commenting on task 1 and 2
4. Show your answers to teacher

Upon successfully grasping the concepts of Boolean expressions to simplify circuits, let us now focus on arithmetic operations and how they can be implemented in Python.

EXPLANATION OF THE ARITHMETIC OPERATIONS

1. Addition (+)

Addition is about combining two or more numbers to get their total. For example, if you have 5 cedis and your mother gives you 10 cedis, you will now have 15 cedis in total. Mathematically, this is represented as 5 + 10 = 15.

Examples:

a. 22 + 3 = 25

b. There are 2 teachers in the staff common room, and 3 more teachers come in. How many teachers are there in the staff common room now? This gives 2 + 3 = ? in mathematics. The answer would be 5 teachers are in the staff common room

Properties to note

a. **The commutative property of addition** means that the order in which you add two numbers does not affect their total.

   In mathematics, it means a + b = b + a. For example, if you are buying Hausa koko and koose. Whether you buy the Hausa koko first and then koose or buy the koose and then Hausa koko, the total cost is the same.

   Examples: 22 + 3 = 5 and 3 + 22 = 25. Similarly, 5 + 7 = 7 + 3 both give the same result as 12.

b. **The associative property of addition** means that when adding three or more numbers, the way the numbers are grouped does not change their total. Mathematically, a + (b + c) = (a + b) + c. For example, you have 30 minutes of study time for Mathematics, 20 minutes for English, and 10 minutes for Computing. Whether you first add the time for Mathematics and English, then add Computing, or you first add English and Computing, the total study time is the same.

   Examples: (30 + 20) + 10 = 30 + (20 + 10) gives 60.

c. **The identity property of addition** means adding zero to any number gives you the same number. Hence, 0 is the identity for addition.

d. Mathematically, $a + 0 = a$. For instance, if the school has 100 computers in the laboratory and no one gives any more computers, the laboratory will still have 100 computers. Thus, 100 + 0 = 100.

2. Subtraction (-)

Subtraction is taking away a number from another. It also means finding the difference between numbers. For instance, if you have 10 bofrots and you give 3 to your sister, you will have 7 bofrots left. Mathematically, it means

10 – 3 = 7.

Examples:

a. 2024 – 224 = 1800.

b. Kofi bought a book for 20 cedis. If he had 50 cedis, he would have 30 cedis left. Thus, 50 – 20 = 30.

Properties to note

a. **The non-commutative** means the order in which the numbers are taken away from each other affects their result. Mathematically, $a - b \neq b - a$.

   Example: 22 − 3 does not give the same result as 3 − 22.

b. **The associative** means that the way the numbers are grouped changes their total. Mathematically, $(a - b) - c \neq a - (b - c)$.Example**:** (30 − 20) − 10 does not give the same result as 30 − (20 − 10).

3. Multiplication (x)

Multiplication means adding a number to itself a certain number of times. That is performing repeated addition.

In computing, "*" is used for multiplication and not "x". Example: If you want to buy 7 projectors but each projector costs GH¢4,500.00, you need to multiply 7 by GH¢4,500.00 to get the total cost of the seven projectors, GH¢31,500.00. Thus, 7 * GH¢4500.00 = GH¢31,500.00.

Examples:

1. 5 * 32 = 160.
2. A farmer by name Imoro from Tumu has 200 sacks. Each sack contains 40 bowls of maize. Mr. Imoro will have 200 * 40 = 8,000 bowls of maize in total.

Properties to note

a. Commutative: Just like addition, the order in which you multiply two numbers does not affect their total. Mathematically, a * b = b * a.

b. Associative: Also, when multiplying three or more numbers, the way the numbers are grouped does not change their total. Mathematically, (a * b) * c = a * (b * c).

c. Identity means multiplying any number by one gives you the same number. Hence, 1 is the identity for multiplication. Example: a * 1 = 1 * a = a. Also, 100 * 1 = 1 * 100 = 100.

d. Distributive combines multiplication and addition. Example: a (b + c) = (a * b) + (a * c). Also, if you want to buy 3 boxes of pens, and each box has both 4 blue pens and 2 black pens. You will need 3*(4+2) = (3*4) + (3*2) = 18.

3. Division (/ or ÷)

Division means splitting (sharing) a number into parts. For instance, if you have 200 cedis and you want to share it among 5 students, you will have to share (split) 200 into 5 parts. Thus 200/5 gives 40 cedis to each student.

Example:

1. 6/3=2.
2. Uncle Akwasi has 24 oranges and wants to share them equally among his four friends: Alberta, Barbara, Nhyira and Fafa. Each friend will get 24/4 = 6 oranges.

Properties to note

a. Non-commutative means that the order in which the numbers are divided affects their result. Mathematically, a/b ≠ b/a (in most cases).

b. Associative means that the way the numbers are grouped before splitting will affect their total. Mathematically, (a/b)/ c ≠ a/(b/c).

c. Identity means any number divided by 1 is that number. Thus a/1=a but not 1/a. Distributive: a / (b * c) = (a / b) * (a / c)

Activity 1.13

Chart on Arithmetic Operation

1. Working with your peers, create a chart on a poster or with computer software, listing each operation with its properties and some examples.
2. Present your work to the class for justification (stay calm and be open to feedback).

PEDMAS

PEMDAS is an acronym that helps you remember the order of operations in arithmetic and algebra, to evaluate expressions 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 in the following order.

1. Parentheses: Simplify expressions inside parentheses first.
2. Exponents: Solve powers and roots next.
3. Multiplication and Division: Work from left to right, performing both operations as they appear in the expression.
4. Addition and Subtraction: Work from left to right, performing both as they appear.

Example

In solving 4+2*(9+18) ÷3^2−5, using PEMDAS, we solve what is in parentheses first, thus 9 + 18 = 27.

Now the expression is 4+2*27÷3^2−5. We then move to the exponent, thus, 3^2 = 9. We now have 4+2*27÷9−5. We move to multiplication, 2*27 = 54, forming a new expression as 4+54÷9−5. We perform division next, 54÷9 = 6.

This forms a new expression as 4 + 6 + 5. We then perform addition and subtraction from left to right.

We now have 4+6 = 10, 10 – 5 = 5.

Hence our final answer is **5**.

Activity 1.14

Application of PEMDAS

1. Solve the following expressions in your groups.
   a. 2-18 ÷4^3−9
   b. 9+6*((9+18) *(3^2)) ÷4^2−3
   c. James Sortortey is building a farm to keep his birds as an agricultural science student. He needs 8 units of wood for each house and 6 units of wood for each barn. If he has 3 houses and 2 barns, how many total units of wood does he need? Write down the steps you take to work out your answer.
   d. Alberta Eyram Geli has 150 minutes to watch two videos on how to mould a trailer as a mechanical engineering student. The first video is 1 hour and 30 minutes long, and the second video is 45 minutes long. How much time does she have left after watching both movies? Write down the steps you take to work out your answer
2. Share your result with the class
3. Make corrections where necessary.

Implementation of Arithmetic Operations in Python

If you do not have a Python IDE installed on your device, click here or *https://www.programiz.com/python-programming/online-compiler/* to write and edit your Python codes online.

Write the following code snippets in your Python IDE.

Activity 1.15

Arithmetic Operation in Python

Copy the following Python codes and run to see the output

1. code for addition

```
a = float(input("Enter your first number: "))
b = float(input("Enter your second number: "))
sum = a + b
print("Sum: ", sum)
```

2. code for subtraction

```
a = float(input("Enter your first number: "))
b = float(input("Enter your second number: "))
difference = a - b
print("Difference: ", difference)
```

3. code for Multiplication

```
a = float(input("Enter your first number: "))
b = float(input("Enter your second number: "))
multiplication = a * b
print("The answer is: ", multiplication)
```

4. code for division

```
a = float(input("Enter your first number: "))
b = float(input("Enter your second number ≠ 0: "))
quotient = a / b
print("The answer is: ", quotient)
```

5. code for exponent

```
a = float(input("Enter your floor number : "))
b = float(input("Enter your exponent number: "))
exponent = a ** b
print("The answer is: ", exponent)
```

Activity 1.16

Python Implementation of Arithmetic Operation

1. Perform the following in groups.
   - **a.** Write a Python program that asks the user for the prices of three items they want to buy, calculates the total cost and gives the average.
   - **b.** Write a program that asks the user for a number and calculates the cube of that number.
   - **c.** Gyampo Anomabo is a corporal in the Ghana Police Service, and he needs to encode a message. Write a Python program that takes any number, multiplies it by 5, subtracts 2, and prints the result. Do the reverse operation.
   - **d.** Mabel Kundi has a savings account with ADB that grows by 5% every year. Write a program that takes her current balance and calculates what the balance will be after 3 years.
   - **e.** Millicent has a savings account with Fidelity Bank that grows by 15% every year. Write a program that takes her current balance and calculates what the balance will be after 5 years.

2. Present your results to your class for discussion.

EXTENDED READING

- Click here or *https://www.javatpoint.com/python-arithmetical-operations* to learn more on arithmetical operations in Python.
- Click here or *https://www.tutorialspoint.com/python/python_arithmetic_operators.htm* to learn how arithmetic operation is done in Python and implementation of arithmetic operators in Python.

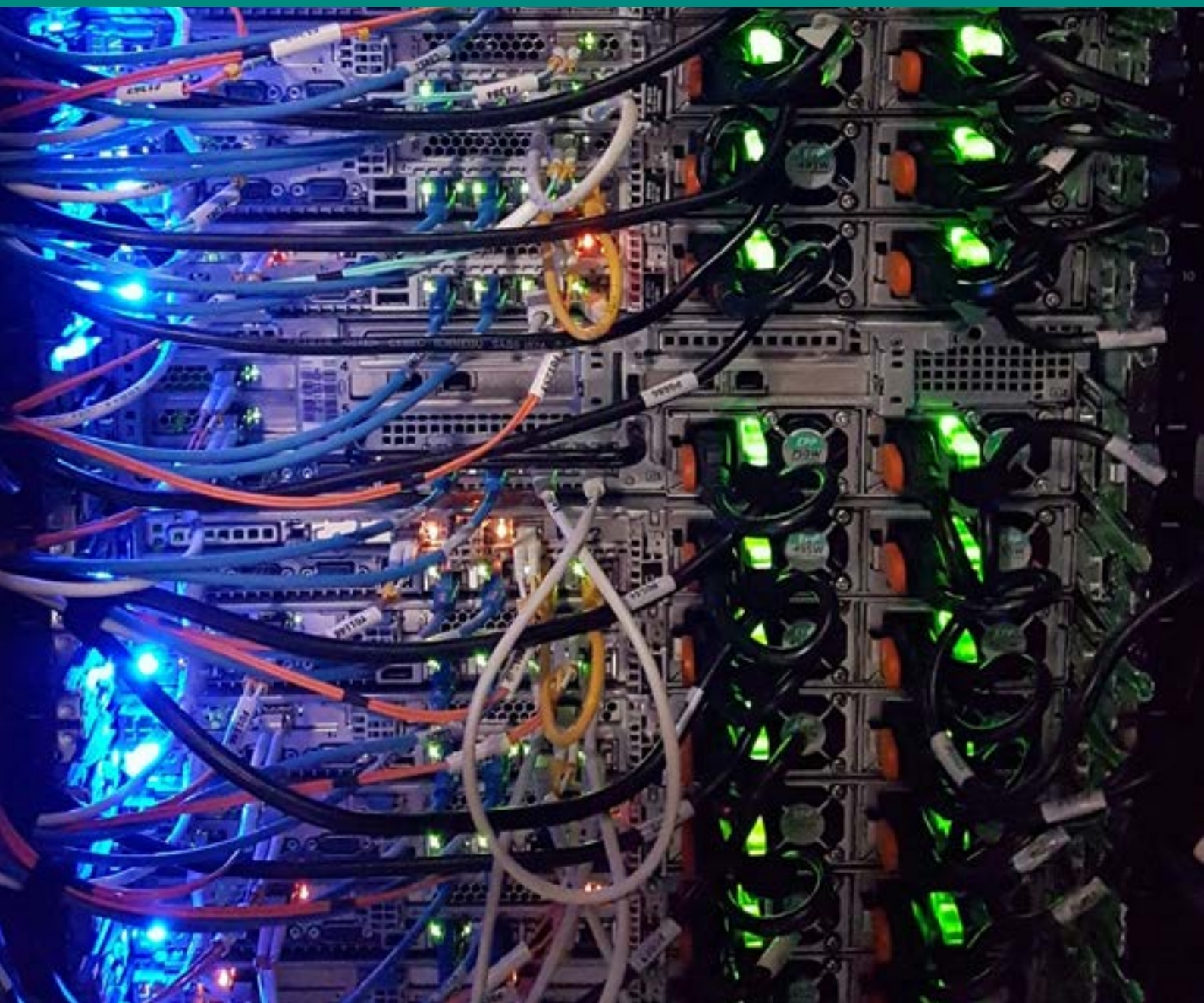
Review Questions

1. What are the three main arithmetic operations?
2. Which of these gives quotient as the result?
   - **A.** Addition
   - **B.** Division
   - **C.** Subtraction
   - **D.** Multiplication
3. Explain how a NAND gate is different from an AND gate. Provide an example using the inputs 1 and 1.
4. Can a NOR gate be used to replace an OR gate in a circuit? Explain why or why not, using examples.

SECTION

2

FUNCTIONALITIES OF HARDWARE, SOFTWARE AND BARCODES



COMPUTER ARCHITECTURE & ORGANISATION

Computer Hardware and Software

INTRODUCTION

Welcome to our next lesson on barcodes, QR codes, and software. In this section, we are going to explore how barcodes and QR codes work and create some of our own. Barcodes and QR codes need software to work properly; we will extend our learning to types of software, their functionality, and how to select the right software for the right work. Barcodes are used everywhere, from the products we buy at the store, to the tickets we use for events and shows.

Key Ideas

- **Barcodes** are a series of black and white lines that store information.
- **QR codes** are a type of barcode that can store much more information than a regular barcode.
- **Software** is a set of instructions or programs that tell a computer what to do. Think of it as the brain of a computer that makes everything work.

BARCODES

A barcode is made up of a series of lines of varying widths that represent numbers and other symbols.

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 to 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. We can also use a scanner to read the barcode on a library book to find out its title and author.

Types of Barcodes

1. **EAN-13 Barcode / International Article Number**: International Article Numbers, formerly European Article Numbers (EAN). EAN is a way to give each product a unique number so that stores can easily keep track of what they sell. It consists of 13 digits. Each digit helps to identify the product and its details. For example, if the code starts with "590," it is from Poland.

   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.

   However, in Ghana, we use the **GHBS-13**, which stands for Ghana Barcode System of Group 13. This system is based on the standard EAN-13 barcode format but has been enhanced to better suit local needs.

   Figure 2.1 shows some examples of IAN barcodes, with a breakdown of each of the sections within the barcode number.

Figure 2.1: IAN Barcode

2. **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 2.2** shows an example of a UPC-A barcode.

Figure 2.2: UPC-A Barcode

Activity 2.1

Distinguishing Barcode

Observe **Figure 2.3** and identify the main differences between an IAN and a UPC barcode. Discuss the differences with your peers.

Figure 2.3: Activity image

3. **ITF-14 Carton & Pallet Barcodes**: ' ITF stands for Interleaved Two of Five. It is a type of barcode used to print a 14 digit product code on shipping cartons and boxes. This is to help companies identify and track goods during shipping. The number it carries is called GTIN-14 (Global Trade Item Number) and was formerly called SCC-14 (Shipping Container Code). Further, GTIN-14 or SCC-14 refers to the number while ITF-14 refers to the barcode symbol on the carton.

   Though ITF-14 is easy for scanners to read on cardboard boxes, works well even with simple printing and helps in quick checking and tracking of goods, it uses more spaces on the box, cannot be be used in supermarket checkout and can only show numbers, not letter ITF-14 Carton Codes are created from EAN-13 and UPC-A barcodes. These types of barcodes are not used for individual products (UPC-A or EAN) but only on cartons containing specific quantities of the products.

Figure 2.4: ITF-14 Carton

4. **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 books often do use ISBNs (simply because they are published titles)."

The ISBN started in 1966 as a 9-digit used by British booksellers. In 1970 it became a 10-digit international code and in 2007 it expanded to 13 digits so that it could fit into the EAN barcode system used in shops. The first three digits, usually 978 or 979 in the ISBN means that the item is a book. The 13-digit ISBN is made up of a prefix, a group number, a publisher number, a title number and a final check digit.

ISBNs are important because they make it easy to identify, order and manage books, however, publishers must obtain new numbers for each format of a book and there are costs involved.

Figure 2.5: ISBN Barcode

5. **ISSN Magazine Barcodes:** An 8-digit International Standard Serial Number (ISSN) uniquely identifies publications such as magazines and journals. An example of an ISSN Barcode is shown in **Figure 2.6**.

Figure 2.6: ISSN Barcode

Activity 2.2

Creating a Barcode

1. Search the internet for a barcode generator
2. Select ISSN in the menu option

3. Type in these numbers: 9789988577292
4. Download or screenshot the barcode
5. Use another device (preferably a smartphone) to search for a barcode reader online
6. Scan the barcode

If your scan result displays EAN and 9789988577292 with other information, then it means your barcode was correctly generated as an EAN-13 (ISBN type), and your scanner has successfully read the encoded number, confirming the barcode is valid and functional.

Activity 2.3

Barcodes in Retail Stores

1. Individually, observe the slide presentation on how barcodes are used in retail stores and write down the importance of the use of barcodes in the presentation
2. In mixed-ability groups of no more than 5, research the types of barcodes and categorise them into retail book and magazine barcodes.
3. Scan for reading resources

In your group, make a presentation on how to create barcodes. (You will be assigned a type of barcode to research.)

Quick Response (QR) Code

QR codes can store more data, and they have found a wide range of uses. In marketing, QR codes can be scanned to reveal product information, promotional content, or direct customers to a website. In the field of education, QR codes can provide quick access to additional resources, such as the video in **Activity 2.4**, to complement learning materials. They are also used in event management, serving as digital tickets that can be scanned for entry. In the entertainment industry, QR codes on movie posters or music albums can link to trailers or samples.

Structure and Features

1. **Structure:** The QR Code is a two-dimensional barcode made up of squares that are black and white; they are arranged in a square grid. The QR code can be scanned with a camera, like those found on smartphones, or you can use specific QR code readers.
2. **Capacity:** QR codes have the capacity to store 7,089 numeric characters or 4,296 alphanumeric characters.
3. **Error Correction:** QR codes have error correction built in; they can be damaged or obscured by up to 30% and still be readable. This is important because QR codes can still work even if they get a bit dirty or they get scratched. You can put QR codes on many different things, like posters, products, or tickets, and they will still work even if a part of them is hidden or damaged. Sometimes, people add logos or pictures to QR codes, the extra data lets them do this without breaking the code.
4. **Variants:** There are various versions of QR codes, and each one has varying capacities for data storage. Version 2, which has the capacity to store 25x25 data modules, is the most widely used version.

Activity 2.4

QR Code Experience

1. Have you ever used a QR code? Maybe at a shop, restaurant, or during an event?
2. What was the purpose of scanning the QR code? Did it lead you to a website, a menu, or maybe a video?
3. If your answers to tasks 1 and 2 are “yes”. Volunteer to share your experience with your classmates.
4. Explain where you found the QR code, how you scanned it and what happened after you scanned it.
5. Emphasise on whether it was easy or difficult to use and if you found it helpful.

Activity 2.5

Exploring QR code

In groups of not more than five

1. Use a phone camera or a QR code scanner app to scan this QR code:

2. Watch the 3-minute video.
3. As a group, in your own words, write a QR code generation guide.

Activity 2.6

Generating a QR Code

1. Working in groups, use your guide to generate a QR code for any educational website of your choice.
2. Scan your QR code to test it is working.
3. Save the QR code.
4. Share your QR code with another group, so that they can visit your chosen website.

It is important to note that there are other ways of generating QR codes by using desktop applications, mobile applications, and other website applications.

Activity 2.7

Application of QR Code

Thinking about how you have experienced QR codes and what has been covered so far in this course. Individually, add 5 additional applications of QR code to the list below.

1. QR codes are used for quick and secure payment processing.
2. QR codes on documents can link to additional resources or digital copies.
3.
4.
5.
6.
7.

Discuss your additional applications with your peers.

What is Software?

Software is a set of instructions computers use to perform their tasks. The main role of a software is to control the hardware and help computer user work easily and effectively. In Ghana, software help teachers in the schools to prepare learner plans, assessments tasks, record your assessment scores or results in the students transcript portal, helps banks to process mobile money transfers and let businesses prepare invoices and manage sales.

Before we discuss the types of softwares, it is good we know that software makes the computer useful by guiding every activity it tasks, from opening a file to running services used in Ghana such as passport application portal, microsite, online transcript portal, mobile money applications, WAEC result systems, school websites, etc. ‘’

Categories of Software and Their Differences

Computers make use of programmes called “software”. Software is a set of instructions or programs that tell a computer what to do. Think of it as the brain of a computer that makes everything work. Without software, a computer would just be a box of metal and plastic that cannot do anything useful.

1. **System Software:** Software is what helps the computer do its job. It’s designed to control the physical parts of the computer (like the keyboard, screen, and memory) and allows the other programs to run smoothly. Examples of system software include operating systems like Windows10 or 11, Red hat, fedora, Ubuntu, Monterey, Ventura, Sonoma, etc.. There are also smaller programs called device drivers that help specific parts of the computer (like a printer or a mouse) work properly. All of these programs make sure the computer can run smoothly and perform tasks like opening apps, browsing the internet, or playing music.
2. **Application Software:** Application software is made to help you do specific things on your computer. For example, Microsoft Office Suite (like Word or Excel) which help you create documents or work with numbers. Web browsers like Chrome and Firefox let you go online to search for information or visit websites and programs like Adobe Photoshop help you edit images. These kinds of software are there to make your work easier, whether it is writing, browsing the internet, or managing your data.

   Examples of application software are:

   i. Word processing application: MS Word, WPS Word, Google Docs

   ii. Spreadsheet application: MS Excel, WPS Spreadsheet, OpenOffice Calc, LibreOffice Spreadsheet, etc.

Activity 2.8

Software in Daily Life Discussion

1. In your group, share your experiences on how a browser or application was helpful or challenging in accessing the YouTube video content on the QR code that you scanned.
2. Reflect on how the computer you used was able to take in the instruction you gave it and produced a result.
3. Write a brief sentence to explain why the computer was able to do this.

Functions of System Software

1. An **operating system (OS)** is like the manager of a computer. It takes care of the physical parts of the computer (hardware) and helps you interact with it. The OS is responsible for running applications and making sure everything works smoothly. It has different parts that handle various tasks such as:

   a. Process management: Controls how programs run on the computer.

   b. I/O device management: Manages devices like keyboards, mice, printers, and monitors.

   c. File management: Organises and stores your files and data.

   d. Network management: Helps the computer connect to the internet and other computers.

   e. Main memory management: Looks after the computer's short-term memory (RAM).

   f. Secondary storage management: Manages long-term storage like your hard drive.

   g. Security management: Protects the computer from unauthorised access.

   All these parts work together to keep the computer running efficiently.

2. **Device drivers** serve as translators between the operating system (OS) and hardware devices, enabling the OS to effectively communicate with components like printers and keyboards.

3. **Firmware i**s a type of software that is embedded directly into hardware devices. It provides low-level control for specific functions of those devices. Firmware for instance can be found in gadgets like printers, cameras, and routers, allowing them to operate correctly and perform their intended tasks without needing a full operating system.

4. **Utilities** are tools that help manage and maintain your computer. They perform important tasks to keep the system running smoothly, like antivirus programs that protect against malware and disk cleanup tools that remove unnecessary files to free up space. These tools ensure your computer stays healthy and efficient.

Functions of Application Software

1. Productivity software helps you work more efficiently by making it easier to create documents, manage spreadsheets, and design presentations. Examples include Microsoft Office, which offers tools like Word for writing, Excel for data organization, and PowerPoint for creating slideshows.
2. Web browsers allow users to access and explore the Internet. Examples include Google Chrome, Mozilla Firefox, and Safari, which help you find information, visit websites, and interact with online content.
3. Media players enable users to play audio and video files. For example, VLC Media Player allows you to easily listen to music and watch videos in various formats.
4. Graphics software lets users create and edit images and designs. An example is Adobe Photoshop, which provides tools for photo editing, graphic design, and digital artwork.

Activity 2.9

Roles of Different Software Types

In your groups;

1. Each group should pick one of these items (OS, device driver, productivity software, user).
2. Create a play demonstrating how the software helps users accomplish any specific task of your choice.
3. Each group will perform their play in front of the class. The class is required to make notes on what they have learnt from each play.
4. Then, engage in a class discussion where everyone can share their responses and learnings from the plays.

EXTENDED READING

- Click *here* or *https://egyankosh.ac.in/bitstream/123456789/33166/1/Unit-4.pdf* to read more on application software.
- Follow this *link* to learn more about how to create and design QR codes *https://www.adobe.com/express/feature/image/qr-code-generator*
- Watch this *video* to learn more about the differences between EAN and ITF-14 Carton & Pallet Barcodes *https://www.youtube.com/watch?v=kGVUlfkpFlw&t=4s*
- Click on this *link* to read more on barcodes *https://barcodesghana.com/wp-content/uploads/sites/96/2020/10/Guide-to-retail-barcodes-Ghana.pdf*

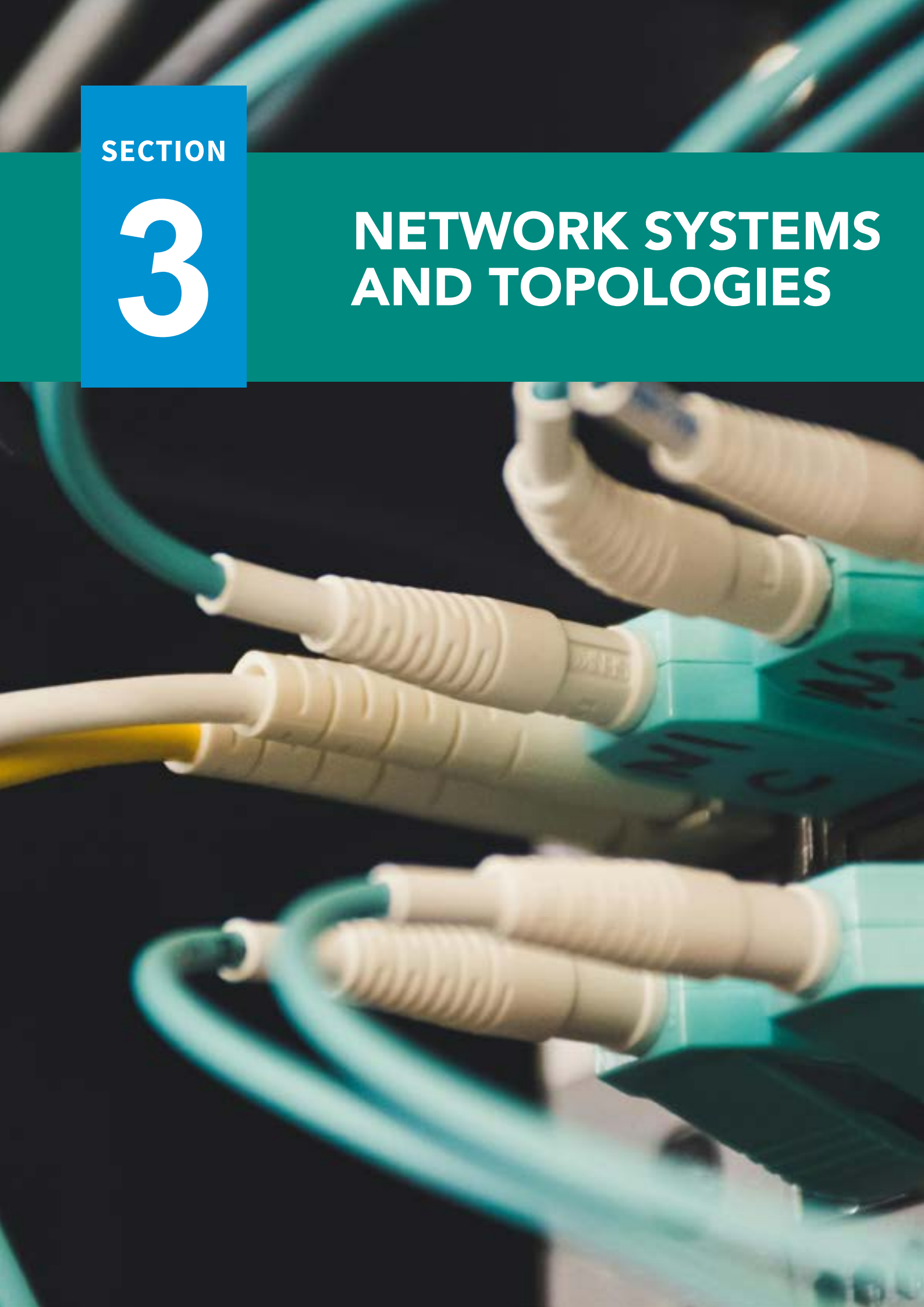
Review Questions

1. What is the primary function of an operating system?
2. Device drivers allow the operating system to communicate with hardware devices. True or False.
3. What does “QR” stand for in a QR code?
4. A QR code can be damaged by up to 50% and it will still scan correctly. **True or False**
5. Discuss the importance of device drivers in computer operation and provide examples of hardware that requires drivers.
6. Explain how a QR code can store different types of data. Give two examples of the types of data it can hold.
7. Explain why it is important to test a QR code after creating it.
8. Kennedy is asking how he can improve his computer’s performance. What utility software would you recommend, and why?
9. How could adding a level of error correction in a QR code be helpful when placing it on a poster outside?
10. Analyse, using real-world examples, the impact of a missing or malfunctioning device driver on a computer system’s overall functionality.
11. Discuss the issues arising from the lack of compatibility of new application software with an existing system.

SECTION

3

NETWORK SYSTEMS AND TOPOLOGIES



COMPUTER ARCHITECTURE & ORGANISATION

Data Communication and Network Systems

INTRODUCTION

As you progress in your learning, we have come to another exciting section of Computer Architecture and Organisation. We will explore data communication and network systems. You will use the knowledge acquired to design and demonstrate the various topologies of a network, networking devices, their practical applications and how they facilitate communication. Also, this section will create the foundation for your further studies and practical applications in networking and data communication.

Key Ideas

- An **Internet Service Provider** (ISP) is a company that provides access to the internet to individuals and organisations.
- **Bandwidth** refers to the amount of data that can be transmitted over a network or internet connection in a given amount of time.
- **Internet Protocol (IP)** is a rule that governs communication over the internet.
- **Network devices** are physical devices used to connect computers and other devices within a network.
- **Subnetting** refers to dividing a larger network into smaller networks called "subnets."
- **Virtual Private Network (VPN)** is a technology that allows people to use the internet by creating a private, encrypted connection.

Activity 3.1

Recap on Computer Networks

Individually, perform the steps below.

1. Create a concept map of a Local Area Network.
2. Show your concept to the teacher and peers.

DESIGNING A LAN

When setting up a Local Area Network (LAN), there are some simple rules (principles) to follow to make sure the network is efficient, scalable, secure and easy to manage.

The principles are explained as follows:

1. **Scalability** is the ability of a system, like a network or a computer, to grow and handle more users or tasks without losing performance. For effective scalability, you have to implement best practices such as:
   a. The use of **modular devices** to easily expand your network. Networks contain a number of different devices; they can be designed to ensure that extra devices can be connected to the network. Extra Switches can be added to allow more wired devices to a network, or extra Wireless Access Points can be added to provide wider coverage or for more devices.
   b. Implementing a **hierarchical design** using the three-layer model (core, distribution, and access) to efficiently organise your network. The core layer serves as the network backbone, enabling high-speed data transfer. The distribution layer links the core and access layers, handling policies and routing, while the access layer connects end devices, such as computers and printers. This structure manages traffic effectively and simplifies adding new devices.
   c. **Planning for future growth**. In the future, there will be an increase in the number of computers and devices in your school or office, and you will add new services like video streaming or online learning without significant changes.
   d. **Incorporating redundancies.** You need to have backup plans to ensure your network is available and even if the network fails, data can take another route.
   e. **The use of redundant links and devices** ensures that the LAN has dual power supplies and backup switches and routers so that the LAN stays on even if one fails.
2. **Flexibility:** In order to build a flexible network, you need to ensure it supports various devices and applications. The network needs to have standard protocols and interfaces that allow communication and compatibility across devices. The network should be designed for both wired and wireless access that accommodates different connection preferences and needs. Also, ensure compatibility with diverse devices. It should support computers, phones, IoT devices, and more.
3. **Performance:** The performance of a network is measured by how the network can handle data loads and provide low latency. In order to maintain network performance and reliability, we need to: (a) use high-speed links (cables) and equipment (switches) such as Gigabit Ethernet or 10 Gigabit Ethernet to support faster data transfer; (b) optimise network traffic with Quality of Service (QoS) policies to manage bandwidth, set priority levels, and minimise delays for critical tasks and use the right cable for the network design. Most homes and small

businesses networks ideally need a minimum of Cat5e or Cat6 cables depending on speed and distance needs.

4. **Security:** It is important to secure your LAN to protect it from unauthorised access. You can do this by using a firewall, using updated antivirus and anti-malware software, strong passwords, and regular software updates.

Activity 3.2

Research on LAN Design

1. Go online.
2. Explore the videos, images, or other interactive tutorials that explain how to use a diagram-creating tool like Microsoft Visio to design layouts for real-world examples of wired LAN.
3. Using the knowledge gotten from (a), design layouts of wired LAN.
4. Present your design to your class.

If you have completed the activity early, you can explore the software – Cisco Packet Tracer, this is a free network planning tool.

Planning

LAN (Local Area Network) planning involves important steps to make sure the network meets both the current and future needs of the organisation. The main steps in planning a LAN include the following

a. **Gather Requirements:** In requirement gathering, you have to specify the number of users, devices, required applications, user needs, performance, security, reliability, scalability, cost constraints, and coverage area of the network. The requirement gathering stage tells the network engineer what the network should be made up of and include.

b. The next step in planning a network is to design network topology. Having the topologies in mind and knowing the appropriate requirements of the network, you need to design the best-fit topology for your network. This is done in order to logically improve performance and security and select an appropriate network topology (e.g., star, bus, ring, mesh) based on scalability, redundancy, cost-effectiveness, and segment.

c. You need to select network equipment and connection media for your network. This includes hardware and connection media.

   i. In **hardware selection**, select network devices (e.g., switches, routers, firewalls, and wireless access points) based on performance requirements, scalability, and budget constraints.

ii. **Connection Media:** Decide between wired or wireless connectivity. If wired, choose cabling such as Shielded Twisted Pair (STP), Unshielded Twisted Pair (UTP), coaxial, or fibre-optic cables and use structured cabling systems for organised, reliable wired networks However, if wireless, choose wireless access points (WAPs) based on coverage, bandwidth, and user capacity, and choose the appropriate Wi-Fi standard (like Wi-Fi 5, 6 or 7). Use omnidirectional or directional antennas as needed and ensure strong security with WPA3 and a guest network. Also, consider a network controller for managing multiple WAPs and scalability for future expansion.

iii. **Address IP Addressing and Subnetting**: Develop a scheme that supports current needs and future growth, taking into consideration segmentation (subnetting).

iv. **Plan for Network Security**: Install security systems such as firewalls, Intrusion Detection Systems (IDS), and Intrusion Prevention Systems (IPS), access control and encryption protocols to secure the network from unauthorised access and potential threats.

Activity 3.3

Planning a LAN to Design

Do this activity in groups by following the steps below.

1. Draft a proposal following the steps on planning a LAN for your school's computer laboratory. You should aim to include the following in your draft proposal:
   a. The requirements
   b. A topology design based on:
      i. Scalability
      ii. Redundancy
      iii. Cost-effectiveness
      iv. Segment
   c. The hardware equipment needed
   d. The connection media needed
   e. The IP Addressing scheme
   f. Network security requirements
2. Do a market survey on your plan.
3. Indicate the budget allocation you considered for the proposal.
4. Present your proposals and justify your design choices to the class.

Activity 3.4

Designing a LAN

1. In your groups, design a LAN for your school's computer laboratory
2. Present your work to the class by explaining your choices and rationale behind your network architecture.
3. Seek clarification on concepts that you do not understand.

You have just planned and designed LAN. Let us now look at the characteristics of CANs (Campus Area Networks), MANs (Metropolitan Area Networks) and WANs (Wide Area Networks) in this learning. Stay tuned and enjoy.

CANs (Campus Area Networks), **MANs** (Metropolitan Area Networks), and **WANs** (Wide Area Networks) networks differ in scope, speed, and management complexity, with CANs being the most localised, MANs covering cities, and WANs spanning wide geographical areas. Understanding their characteristics will help you identify the most appropriate network type for specific geographic, organisational, and connectivity needs.

CHARACTERISTICS OF CANS, MANS, AND WANS

It is vital to identify the most appropriate network type so that you can design the appropriate needs of the organisation. Let us look at what makes each network type different from another.

1. **Campus Area Networks (CANs)** are fast networks that use cables like Ethernet or fibre optics to connect several Local Area Networks (LANs) within a small area, such as a school campus. They are usually controlled by one organisation.
2. **Metropolitan Area Networks (MANs)** are a bit slower than LANs but faster than WANs. They connect several LANs within a city. Multiple organisations can manage them, making them suitable for businesses or institutions that operate in the same area.
3. **Wide Area Networks (WANs)** often use rented lines or satellite connections. They link different CANs, MANs, and LANs over large distances, such as across a country. These networks usually involve many service providers and organisations working together.

Components of CAN/MAN/WAN design

1. The key components in designing **CAN/MAN/WAN design** include routers, switches, modems, multiplexers and leased lines. These components work

together to create efficient, secure, and reliable networks, whether for a campus, a city, or across large distances. Knowing these components will enable you to select the appropriate device in designing your CAN/MAN/WAN network.

a. Routers help to send packets (data) from one network to another by figuring out the best route (path) to transport the data.

b. Switches connect individual devices, such as computers and printers, within a network and help send data to the right device using their MAC addresses. They help in managing data flow within LANs, which helps in improving the performance of CANs and MANs.

c. 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.

d. A multiplexer combines several signals into one signal and sends all these over a single medium, which are separated at the final destination. It is good for WANs and MANs as it helps to improve the usage of bandwidth by enabling more data to be sent at a time.

e. Leased lines dedicated and private communication lines provided by a service provider. They offer fast, reliable connections which can be used in WANs for connecting long distance locations such as company offices spread across a geographical area.

2. The security considerations for CAN/MAN/WAN design include the following:

a. Encryption changes data into an unreadable format (secret code) that can only be decrypted (read) by authorised people. However, this keeps the data confidential (private) and secure while it is being sent across the network, protecting it from unauthorised users who should not see it.

b. 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. Authentication verifies or checks if the person or device trying to access the network is who they claim to be. This helps to ensures that only authorised (approved or verified) people can access the network, keeping unauthorised users out.

c. 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. Access control sets rules for who can access different parts of the network and what they can do. This is done using access levels based on roles a user assigned or performs in the system. This helps in protecting data by making sure people can only see and use the information they are supposed to, keeping the network secure and data safe.

Activity 3.5

Designing a CAN, MAN and WAN

In your group, design a network for a school, a city library system or for GCB bank (nationwide).

1. Create a basic CAN, MAN, or WAN design for your chosen environment.
2. Select the necessary components and identify connection types for your network.
3. Choose security feature(s) that are vital to your network.
4. Present your group design and explains your choice and how each component fits your chosen network type (architecture) and keeps it secured.

NETWORK CONNECTIVITY

Network connectivity refers to the ability of devices and other systems to communicate and interact with each other over a network. It is the foundation of modern computing and communication, enabling the exchange of data, information and resources between different devices and locations.

Network connectivity can be established through various technologies and protocols, such as wired connections using Ethernet cables, wireless connections such as Wi-Fi or cellular networks, and optical fibre connections for high-speed data transmission.

Activity 3.6

Writing Presentation Summary

Individually, write a brief summary of how each of the following is used to connect to the internet

1. Ethernet cables
2. Wi-Fi
3. Cellular

Let us now explore the types of network connections and their characteristics.

Types of Network Connections and their Characteristics

1. Wired Connections

a. Ethernet

Characteristics:

i. Uses twisted pair cables for physical connectivity.

ii. Provides high-speed and stable data transfer rates.

iii. Widely used in homes, offices, and data centres.

iv. Offers reliable and secure data transmission.

v. Easy to set up and maintain.

b. Token Ring

Characteristics:

i. Employs a token-passing mechanism for data communication.

ii. Utilises a ring topology where each node has two connections.

iii. Provides deterministic data transfer with predictable performance.

iV. Often used in industrial automation and manufacturing environments.

V. Relatively complex to set up and manage.

2. Wireless Connections

a. Wi-Fi

Characteristics:

i. Utilises radio waves to establish wireless connectivity.

ii. Offers flexibility and mobility for devices without physical cables.

iii. Supports various standards, including 802.11a/b/g/n/ac/ax.

iv. Provides varying speeds and ranges depending on the standard.

v. Widely used in homes, offices, public spaces, and mobile devices.

b. Bluetooth

Characteristics:

i. Employs short-range radio waves for wireless communication.

ii. Designed for data transfer and device connectivity over short distances.

iii. Widely used in wireless headsets, speakers, and IoT devices.

iV. Offers low power consumption and easy pairing capabilities.

c. Cellular

Characteristics:

i. Utilises cellular networks to provide wireless connectivity.

ii. Enables mobile devices to access the internet and make calls.

iii. Supports different technologies like 3G, 4G, and 5G.

iv. Offers varying speeds and coverage depending on the network.

v. Widely used in smartphones, tablets, and laptops.

3. Optical Fibre Connections

Characteristics:

a. Employs glass or plastic fibres to transmit data using light pulses.

b. Offers extremely high bandwidth and low latency.

c. Immune to electromagnetic interference and provides high security.

d. Ideal for long-distance data transmission, such as undersea cables.

e. It is relatively expensive to install and maintain.

4. Satellite Connections

Characteristics:

a. Utilises satellites in geostationary orbit for data communication.

b. Provides connectivity to remote areas lacking terrestrial infrastructure.

c. Offers global reach and can cover vast geographical regions.

d. Supports various applications, including internet access, telephony, and broadcasting.

e. Latency is typically higher compared to other types of connections.

Activity 3.7

Research work

Working with your peers,

1. Research the meaning, roles and typical costs of the following in the network infrastructure:
   a. Routers
   b. Switches
   c. Wireless Access Points
2. Create a mind map detailing the concepts of network connectivity that you have researched

3. Compare your mind map and findings with the content below to add to your learning.

Role of Network Devices in Establishing and Managing Internet Connections

Network devices play a critical role in establishing and managing internet connections. They enable the transmission of data packets between different networks and provide various functions such as routing, switching, and security. Here are some key network devices and their roles:

1. Routers

A router is a device that connects different networks together. Think of it as a traffic director for data, helping to send information where it needs to go.

Roles:

a. Connecting Networks: Routers connect your home or school network to the internet and can also connect multiple local networks (like different classrooms or offices).

b. Data Routing: When you send or receive data (like loading a website), the router determines the best path for that data to travel. It reads the destination address of the data and forwards it accordingly.

c. Security: Many routers have built-in security features like firewalls that help protect your network from unwanted access.

d. Assigning IP Addresses: Routers typically are responsible for assigning unique IP addresses to devices on the network, allowing them to communicate with each other.

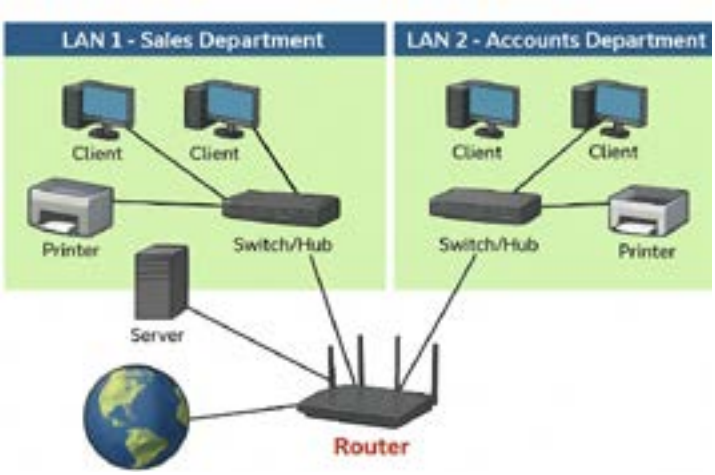


Figure 3.1: Router Connectivity

2. Switches

A switch is a device that connects multiple devices within the same network, such as computers, printers, and servers.

Roles:

a. **Data Forwarding:** Switches receive data packets from one device and forward them to the correct destination device within the same local area network (LAN). This is like a post office sorting letters to send them to the right addresses.

b. **Creating Networks:** Switches help create a local network by connecting several devices, allowing them to communicate efficiently.

c. **Improving Performance:** By directing data only to the intended device rather than broadcasting it to all devices on the network, switches reduce unnecessary traffic and improve overall network performance.

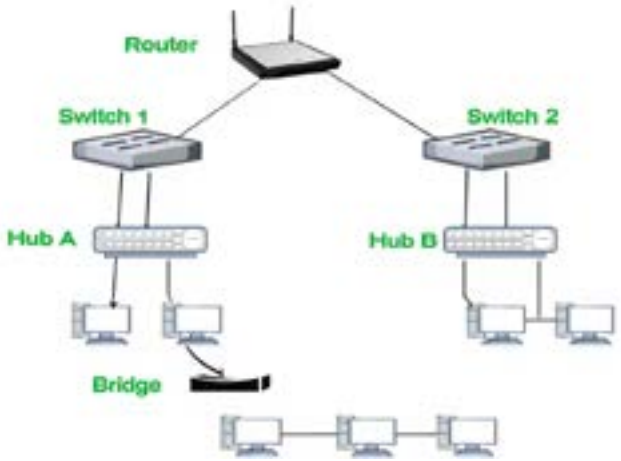


Figure 3.2: Switch Connectivity

3. Wireless Access Points

A wireless access point (AP) is a device that allows wireless devices (like smartphones, tablets, and laptops) to connect to a wired network using Wi-Fi.

Roles:

a. **Wireless Connectivity:** Access points extend the range of a wired network by providing wireless access. This means you can connect your devices without needing cables.

b. **Network Expansion:** They can be used to expand the coverage area of a Wi-Fi network, allowing more devices to connect even if they are far from the main router.

c. **Managing Connections:** Access points can manage multiple connections at once, ensuring that many users can access the internet simultaneously without slowing down.

Internet Protocols (TCP/IP)

The Internet Protocol Suite, also known as TCP/IP, is a set of communication protocols that enables data to be transmitted between computers over the Internet. It is the foundation of the Internet and other computer networks, and it defines the rules and formats for how data is transmitted and received.

1. **Transmission Control Protocol (TCP)**: TCP makes sure that data (such as messages or files) is sent and received correctly. When you send something over the internet, TCP breaks it into smaller pieces called "packets," keeping track of these packets to make sure they arrive in the right order. If any packets are lost or damaged during transmission, TCP will notice and ask for them to be sent again.
2. **Internet Protocol (IP)**: IP is responsible for addressing and routing the data packets. Every device connected to the internet has a unique address called an IP address. This is like your home address, but for computers. IP directs the packets to their destination based on their IP address, ensuring they reach the right computer.

Importance of TCP/IP

a. All computers connected to the internet understand TCP/IP, which allows different types of devices (like laptops, smartphones, and tablets) to communicate with each other, even if they are made by different companies.

b. Thanks to TCP, you can trust that your emails, videos, and games will arrive safely and in order.

c. Protocols like SSL (Secure Sockets Layer) and its successor TLS (Transport Layer Security) encrypt data transmitted over the internet. They ensure that sensitive information, such as credit card numbers and passwords, is protected from eavesdropping and interception.

d. Application layer protocols like HTTP (Hypertext Transfer Protocol) and FTP (File Transfer Protocol) define how applications communicate with each other. HTTP enables the transfer of web pages and other resources over the internet, while FTP allows for the transfer of files between remote computers.

e. Internet protocols are designed to handle the enormous volume of data and traffic that flows through the internet daily. They provide mechanisms for load balancing, congestion control, and error recovery, ensuring that data reaches its destination reliably and efficiently, even in challenging network conditions.

Network Addressing

1. **IPv4** stands for Internet Protocol version 4. It is a way to identify devices on the internet using a unique address. Think of it like your home address, which tells people where you live.

   It uses a 32-bit address, which means it has a total of around 4.3 billion unique addresses. These addresses are written in a format called dotted decimal notation,

which looks like this: **192.168.1.1**. Each number is between 0 and 255, separated by dots.

2. **IPv6** stands for Internet Protocol version 6. It was created because we were running out of IPv4 addresses due to the increasing number of devices connected to the internet, like smartphones, tablets, and computers.

   IPv6 uses a 128-bit address, which allows for a nearly unlimited number of unique addresses—about 340 undecillions (that is 340 followed by 36 zeros!). This is written in a different format called hexadecimal, which looks like this: **2001:0db8:85a3:0000:0000:8a2e:0370:7334**. It uses numbers and letters (A-F) and is separated by colons.

   Importance of IPv6

   a. Because IPv4 has limited addresses, we needed IPv6 to ensure that every device can have its own unique address without running out.

   b. IPv6 includes stronger security measures built right into it, making it harder for hackers to attack devices.

   c. With so many more addresses available, data can travel more efficiently across the internet.

Authentication and Security Mechanisms

When you use the internet, it is important to keep your information safe and secure. There are several ways to do this and here are some key mechanisms that help protect your online activities.

1. Protected Access (WPA/WPA2/WPA3 Wi-Fi)

WPA (Wi-Fi Protected Access), WPA2 and WPA3 are security methods used to protect wireless networks, like the Wi-Fi in your home or school.

How it works

a. **Encryption:** This means that any data sent over the Wi-Fi is scrambled so that only the people with the right password can read it. Imagine sending a secret message that only your friend can understand!

b. **Preventing Unauthorised Access:** WPA/WPA2/WPA3 makes sure that only people who know the password can connect to the Wi-Fi. This keeps strangers from using your internet connection or seeing what you're doing online.

2. Virtual Private Networks (VPNs)

A VPN is like a secure tunnel for your internet connection. It helps keep your online activities private.

How it works

a. **Safe Connections:** When you use a VPN, all the information you send and receive is encrypted, which means it's turned into a code that no one else can read.

b. **Privacy:** A VPN hides your IP address (which is like your internet address), making it harder for websites and others to track what you're doing online. This is especially useful if you're using public Wi-Fi, like in a café or library.

3. SSL/TLS (Secure Sockets Layer/Transport Layer Security)

SSL and TLS are technologies that protect the data sent between your computer (or phone) and websites you visit.

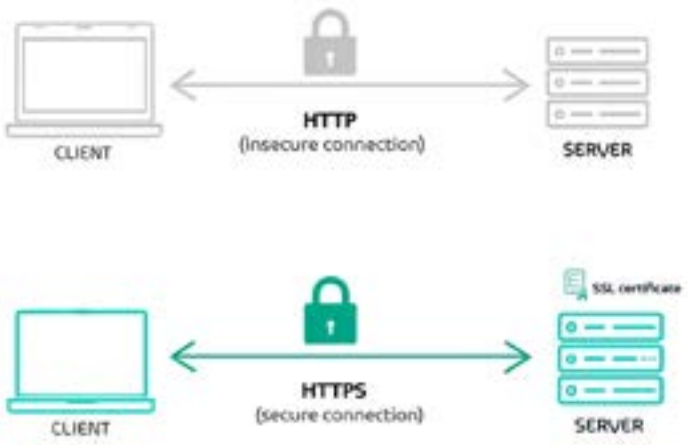


Figure 3.3: SSL

How it works

a. **Encryption of Data:** When you visit a secure website (you will see "https://" at the beginning of the web address), SSL/TLS encrypts the data being sent. This means that even if someone tries to intercept it, they will not be able to read it.

b. **Protecting Sensitive Information:** This is especially important when you are entering personal information, like passwords or credit card numbers. SSL/TLS ensures that this information stays private and safe from hackers.

Activity 3.8

Discussion

Discuss your understanding of authentication and security when using the internet with your peers and your teacher.

CONFIGURATION AND SETUP PROCESS FOR NETWORK CONNECTIONS

1. **Steps to Configure IP Addressing**

   Configuring (setting) an IP address on a device on the network can be either manually which is called **Static IP Assignment** or dynamically, called **Dynamic IP Assignment**. These two assignment processes are discussed below.

   a. **Static IP Assignment:** Under static assignment, we manually assign a fixed IP address to a device. Most devices that we assign IP addresses to manually are servers, printers, and network devices. On these devices, the IP addresses do not change and that helps to ensure stability, reliability and security.

   b. **Dynamic IP Assignment:** To assign dynamic IP addresses automatically, we use DHCP, which stands for Dynamic Host Configuration Protocol. This protocol assigns the IP address, subnet masks, and default gateway. This DHCP is usually used by the router for the assignment. DHCP usually assigns IP addresses automatically to client service devices like laptops and smartphones.

Activity 3.9

Configuring Static IP Assignment on Windows

Individually, watch the demonstration of the teacher or follow the steps below to set up **Static IP Assignment** on a Windows device and answer the questions that follow in your notebook.

1. Type “View Network Connection.”
2. Click on “Change adapter settings.”
3. Right-click on the network connection type and select “Properties.”
4. Double click “Internet Protocol Version 4 (TCP/IPv4)” and click “Properties.”
5. Choose “Use the following IP address” and enter the following as IP address [192.168.1.10], subnet mask [255.255.255.0], and default gateway [192.168.1.1].

Note

This is a sample IP address, its subnet masks, and default gateway. Feel free to modify them for your specific network requirements. This **MUST** align with the network addressing in place on the network, if it does not match then your device will not work on that network.

Self- Assessment

1. What have you observed generally about what just did?
2. What were some challenges you faced while going through the steps?
3. Why do you think it made you face such challenges?
4. What were the solutions to the challenges you faced?
5. What other way can you assign a static IP address better on Windows?
6. Share your response with your peers in class.

2. **DNS (Domain Name System)**: DNS converts domain names into IP addresses. Website address such as *www.google.com* is converted into IP address: 8.8.8.8. The protocol responsible for such conversion is DNS. DNS Servers are provided by ISPs, public DNS providers (e.g., Google DNS: 8.8.8.8), or configured locally.

   How do we configure the DNS? Stay tuned as we practice it in **Activity 3.10**.

Activity 3.10

Steps to Configure DNS

In your groups, watch the teacher's demonstration or follow the steps below religiously to configure DNS.

1. Open Settings
2. Select Network and Internet
3. Select Change Adapter Options
4. Right-click the network you are connected to and select Properties
5. Choose Internet Protocol Version 4
6. Select Properties > Use the following DNS server addresses:
7. Choose a DNS address for IPv4
8. Select OK
9. You can also use Google DNS as your primary or secondary DNS server. The Google Public DNS IPv4 addresses are 8.8.8.8 and 8.8.4.4. The Google Public DNS IPv6 addresses are 2001:4860:4860::8888 and 2001:4860:4860::8844.

You have undergone training on how to configure DNS, which is a vital skill for at IT technician, as well as further studies. Also, you have just enhanced your troubleshooting or problem-solving skills.

Getting on, let us continue with our configuration learning by learning how to configure Wi-Fi on a laptop/PC.

Activity 3.11

Configuring Wi-Fi on a Laptop/PC

In your groups, set up Wi-Fi on your laptop/PC using the steps below.

1. Open Network Settings:
2. On Windows: Click on the Wi-Fi icon in the taskbar and select "Network & Internet settings."
3. On Mac: Click on the Wi-Fi icon in the menu bar and select "Open Network Preferences" or "Wi-Fi Settings."
4. Select Wi-Fi Network:
5. Click on "Wi-Fi" or "Wireless" settings.
6. Select the desired Wi-Fi network from the list of available networks.
7. Enter Password:
8. Enter the Wi-Fi network password when prompted.
9. Click "Connect" or "Join."

Viola, you have successfully configured Wi-Fi on a laptop/PC. Thus great. Let us move to the next learning to configure Wi-Fi on a smartphone.

Activity 3.12

Configuring Wi-Fi on a Smartphone

1. Open Wi-Fi Settings:
2. On Android: Go to "Settings" > "Network & Internet" > "Wi-Fi."
3. On iPhone: Go to "Settings" > "Wi-Fi."
4. Select Wi-Fi Network:
5. Choose the desired Wi-Fi network from the list of available networks.
6. Enter Password:
7. Enter the Wi-Fi network password when prompted.
8. Tap "Connect" or "Join."

Incredibly, you have successfully configured Wi-Fi on a smartphone.

Let us now set up cellular data connections to access the internet.

Activity 3.13

Setting Up Cellular Data Connections

1. Insert SIM Card:

2. Ensure a valid SIM card is inserted into the phone.
3. Open Mobile Data Settings:
4. On Android: Go to "Settings" > "Network & Internet" > "Mobile Network."
5. On iPhone: Go to "Settings" > "Cellular" or "Mobile Data."
6. Enable Mobile Data:
7. Toggle the "Mobile Data" or "Cellular Data" switch to "On."
8. 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."

9. Enter the APN settings provided by the cellular carrier.
10. Verify Connection: Ensure the device shows a successful Ethernet connection.
11. Open a web browser to check internet connectivity.

Finally, under Configuration and Setup Process for Network Connections, let us configure Ethernet on a laptop/PC for easy data communication.

Activity 3.14

Configuring Ethernet on a Laptop/PC

In your groups, follow the steps and provide answers to the questions that follow in your notebook.

1. Connect Ethernet cable
2. Plug one end of the Ethernet cable into the device's Ethernet port.
3. Plug the other end into the router or switch.
4. Open Network Settings:
5. On Windows: Right-click on the network icon in the taskbar and select "Open Network & Internet settings."
6. On Mac: Click on the Apple menu, select "System Preferences," and choose "Network."
7. Configure IP Settings (if necessary):
8. 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.
9. On Mac: Select "Ethernet," click "Advanced," and configure TCP/IP settings.

We are now going to learn about Internet service providers (ISPs) and some issues and troubleshooting techniques related to internet connectivity. Let us get started!

Activity 3.15

Introduction to ISPs

Individually, write in your jotter,

1. What an Internet Service Provider (ISP) is?
2. Why might people need ISPs?
3. How do ISPs help individuals connect to the internet?
4. What ISPs you are familiar with?

Wow! We have just discussed ISPs in **Activity 3.15**. Let us learn it in detail.

INTERNET SERVICE PROVIDERS (ISPS)

Internet Service Providers (ISPs) are companies that connect people, businesses, and organisations to the internet, allowing them to go online and access websites, social media, emails, and more.

Functions of ISPs

1. **Internet Access**: ISPs set up essential network devices such as cables, routers, and data centres, which enable the user to have access to the internet and communicate with others. These devices enable users to have internet access for their tasks.
2. **Bandwidth Management**: ISPs have a function to control the flow of data to make sure internet connections are fast and stable for all users.
3. **Customer Support**: ISPs give technical support, such as troubleshooting, when users face connectivity problems. They help users solve problems relating to the internet connection.
4. **Service Plans:** ISPs provide different internet packages with various speeds, data limits, and prices to suit the needs of different customers. These plans make it easy for customers to choose what they can afford and what they need. It takes care of a variety of budgets.
5. **Security:** ISPs protect users from online threats by providing security measures such as firewalls, antivirus, and spam filters. This secures the users online and keeps them safe in the network.

Activity 3.16

Functions of ISPs

In your groups,

1. Go online and search on the assigned function of ISPs: Internet Access, Bandwidth Management, Customer Support, Service Plans, or Security.
2. Create a simple poster that visually explains the function.
3. Create a simple presentation using flashcards to explain your assigned function, including real-world examples and their benefits.
4. Present your findings to the class.

Types of Internet Access Technologies

1. **Dial-Up Internet**: It is one of the oldest ways to connect to the internet. It uses a regular phone line so that you cannot make calls while using it. Dial-up is very slow compared to modern internet connections.

Figure 3.4: Dial-up

2. **Broadband Internet:** It has a high-speed internet that is much faster than dial-up and is always connected. With broadband, you can quickly browse the web, stream videos, and download files without waiting too long.

Figure 3:5: Broadband

There are different types of broadband internet. These are:

a. DSL (Digital Subscriber Line)

b. Cable Internet

c. Fibre-optic Internet

d. Satellite Internet

3. **Wireless Internet**: uses radio signals instead of cables to provide internet to users. It either uses **cellular networks** such as MTN, Telecel, or AT and offers internet through data bundles for mobile devices or Wi-Fi hotspots.

Activity 3.17

Types of Internet Access Technologies

1. Go online, research the differences between broadband and dial up.
2. Prepare a presentation to share with the class

Wireless Internet Types

3G, 4G, and 5G are types of mobile internet that let you use the internet on devices like smartphones and tablets. They work by connecting your device to cell towers using radio waves, which link to the internet.

Fixed wireless internet is a high-speed internet option for specific places, like homes or businesses. It uses wireless signals to connect to the internet, but it is meant for one location—it does not move around with you. For example, a home may use fixed wireless internet with an antenna that connects to a nearby.

Mobile and flexible internet allows you to stay connected while moving around, such as when using mobile data on your phone. It is provided through cellular networks (3G, 4G, or 5G) and works wherever there is network coverage so that you can use it on the go—whether walking, travelling, or in a vehicle.

Activity 3.18

Internet Common Issues and Troubleshooting Techniques

In your groups,

1. Research and write at most five 5 troubleshooting techniques each for these common internet issues
2. Show your findings to the teacher for approval

EXTENDED READING

- Cisco Networking Academy. (2021). Introduction to networks. Cisco. Retrieved from *https://www.netacad.com*
- Cisco Networking Basics (Episode 1): Network Types and Sizes (LAN, MAN, WAN)
- Cisco Systems. (2020). Cisco networking essentials. Cisco Press. Available at Cisco
- Click here or *https://www.youtube.com/watch?v=YdM7yBrtAck* to learn about creating WAN, LAN, CAN using student packet tracer.
- Click *https://www.youtube.com/watch?v=D4OxpdjfjeI* to watch how to use a diagram-creating tool like Microsoft Visio to design layouts for real-world examples of wired LAN.
- Click on link to access more content on TCP/IP: *https://www.cloudns.net/blog/tcp-transmission-control-protocol-what-is-it-and-how-does-it-work/*
- Google. (n.d.). *Networking and IT security*. In *Google IT Support Professional Certificate*. Coursera. Retrieved from *https://www.coursera.org/professional-certificates/google-it-support*
- How-To Geek. (n.d.). *IP addressing and subnetting for beginners*. Retrieved from *https://www.howtogeek.com/*

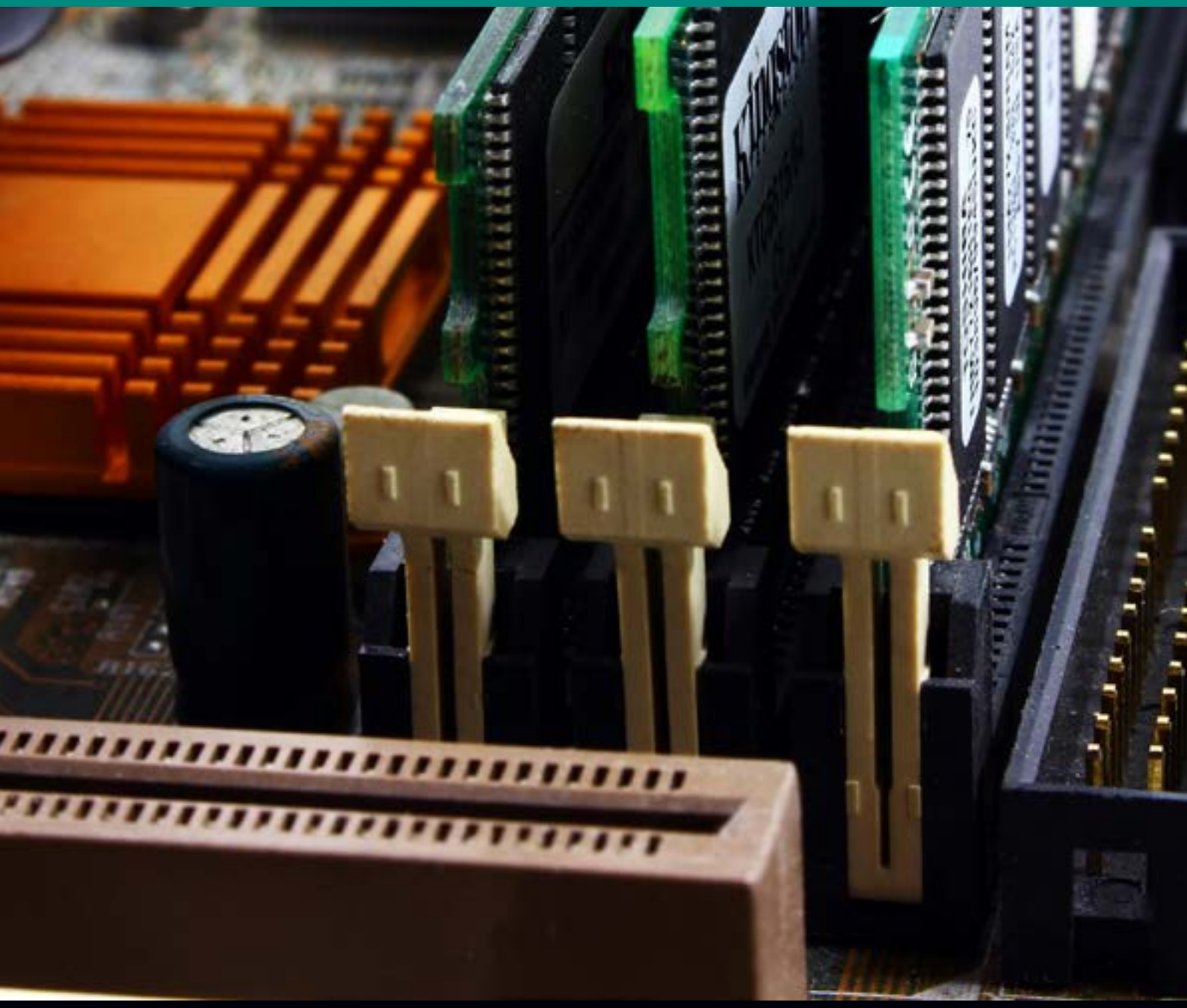
Review Questions

1. Name one scenario where a static IP address would be more appropriate than a dynamic IP.
2. List two troubleshooting steps for a slow internet connection.
3. Name two types of broadband internet.
4. Define TCP/IP and its primary function.
5. List the main differences between IPv4 and IPv6 addresses.
6. If your device connected to the network cannot reach the internet, what could be one possible reason related to DNS?
7. Discuss what could be a reason for a device failing to connect to a Wi-Fi network even after entering the correct password.
8. What possible connection methods are there for connecting to wireless internet?
9. What should you do if there is no internet connection at all?
10. Discuss the importance of SSL/TLS in secure data transmission.
11. Discuss the main difference between HTTP and HTTPS
12. Why might a business use both wired and wireless connection media in a LAN?
13. What are the advantages of using fibre-optic cables in a LAN?
14. Hellen Gorden is setting up a small office network with 15 devices. Which one would you advise him to use: static or dynamic IP addressing, and why?
15. Explain the differences between 3G, 4G and 5G.
16. Explain how VPNs provide security for remote connections.
17. Why is it important to select an appropriate network topology, and what factors should be considered?
18. If your school's network grows beyond its original capacity, what changes will you recommend accommodating more devices?
19. If Ziavi SHTS of Education Winneba uses a MAN to connect its three campuses in Winneba, how will they implement access control to maintain student and staff privacy?
20. How are mobile and flexible internet different from fixed wireless?
21. Design a secure network setup for a small business, incorporating various internet protocols and security mechanisms.
22. Discuss how organisations can prepare their WANs for the integration of IoT devices in the coming years, taking into consideration both efficiency and security

SECTION

4

MAIN MEMORY & DATA STRUCTURES



COMPUTATIONAL THINKING (PROGRAMMING LOGIC).

Algorithm and Data Structure

INTRODUCTION

In this section on Computational Thinking (Programming Logic), we are diving into Algorithms and Data Structures. You will learn how to organise and store data in ways that make it easier to manage and use. We will look at different methods of storing data, understand how they work, and see how they help computers handle information smoothly and efficiently. This foundation will prepare you for more advanced topics in programming and data management, as well as real-world applications in technology.

Key Ideas

- **Data structure** is a special way of organizing and storing data so that it can be used efficiently.
- **Data type** is a kind of label for the different types of information that computers work with.
- **Graph** is a structure used to show connections between items.
- **Hash function** is a tool that changes information (like a name or a number) into a different, shorter form.
- **Hash table** is a special data structure that stores data in a way that allows for quick retrieval.
- **RAM** is a type of computer memory that temporarily stores information while the computer is working on it.

MAIN MEMORY (RAM—RANDOM ACCESS MEMORY)

In year one, you briefly learnt about RAM. Let us recap on the RAM lesson by performing **Activity 4.1** below.

Activity 4.1

Recap on the RAM Lesson

1. Individually, answer the following questions in your notebook.
   a. What is RAM?
   b. Why is RAM an important part of a computer?
   c. How does RAM differ from storage devices such as Hard Disk Drive or Solid-State Drive?
   d. What are the similarities and differences between RAM and your notebook, when we compare RAM with your notebook where you write your notes temporarily?
   e. What unique role does RAM play in a computer system compared to other storage devices?
   f. Given the choice for you to upgrade RAM or CPU, which would you upgrade? Explain your chosen option, taking functions of RAM into consideration.
2. Present your responses to the class for feedback.

Features of memory (RAM)

RAM is like the computer's short-term memory. It is where the computer temporarily stores the data and programs it is currently working on. This makes it easy for the processor to quickly access the information it needs. However, once the computer is turned off, everything in the computers RAM is cleared and disappears. Features of RAM include its capacity, speed and volatility.

1. **Capacity**: RAM is measured in gigabytes (GB), commonly ranging from 2GB to 32GB or more in computers, smartphones, and other devices. The amount of RAM a computer has affects its ability to run multiple programs smoothly. More RAM allows for better performance when handling several tasks at once.
2. **Speed**: RAM speed, measured in megahertz (MHz), determines how quickly data can be read from or written to RAM. Higher speeds improve the computer's ability to process information rapidly. There are different types of RAM, like DRAM and SRAM, each with varying speeds that impact performance.
3. **Volatility**: RAM is considered to be volatile memory, meaning it only keeps data as long as the computer is on. Once the power is off, all information stored in RAM is lost. This makes it different from permanent storage like hard drives or SSDs, which retain data even when the device is turned off.

Types of RAM

The types of RAM include the following:

1. **DDR4 (Double Data Rate 4)** is a type of RAM that is commonly used in modern computers. It is faster and more efficient than previous versions like DDR3, which helps improve performance, especially in tasks requiring high-speed data transfer.
2. **DDR5 (Double Data Rate 4/5)** is the latest version in the DDR series and is even faster than DDR4. It provides higher data transfer rates and improved energy efficiency, making it the best choice for advanced computing tasks, gaming, and high-performance applications.
3. **SDRAM (Synchronous Dynamic RAM)** is a type of RAM that operates in sync with the computer's processor allowing for faster data access compared to older types of RAM. DDR4 and DDR5 are actually types of SDRAM, but SDRAM can also refer to earlier versions used in older computers.

Activity 4.2

Features and types of (RAM)

1. Organise yourselves into groups of no more than five. In your groups, use the internet or textbooks to research on
   a. Any one feature of RAM: Capacity, Speed or Volatility
   b. Any one type of RAM (DDR4, DDR5, SDRAM).
   c. Which type of RAM is most effective for the following tasks: i) gaming ii) server use iii) everyday computing. Provide evidence to support your answer.
   d. Compile your findings in a presentation format (digital or paper-based). Include pictures and images to make your presentation more interesting.
2. Present this to your class for discussion and feedback.

Extension Activity 1

The Future of RAM Technology

1. Use the internet and find out recent innovations or future developments in RAM technology.
2. Identify potential uses for faster or higher capacity RAM and the new types of applications this technology could support.
3. Indicate how developments in RAM technology could change how we use computers or devices in the future.
4. Present your findings to the class.

Data Structures

Introduction to Data Types

Just like we have different types of food (fruits, vegetables, and grains), computers have data types to organise information. Examples include whole numbers (integers), words or text (strings), and true/false answers (Booleans). Computers use these data types to organise, store and handle the information you give them.

a. **Integers** are whole numbers that do not have any fractions or decimals. They can be positive, negative, or even zero. For example, the number of students in a class or the score in a game are integers. Like -10, 0 and 40.

b. **Floats** unlike integers, are numbers that have a decimal point, like 0.142 or 567.14. They are used to represent measurements that require accuracy, such as temperature, height or weight. These numbers can be very large or very small.

c. **Characters** are single symbols, letters or digits that we see or use in our everyday typing. These are used to create words and sentences. For example: 'D', 'm', '2', and '@' are all samples of characters that help us form the text we see on the screen.

d. **Pointers** are special types of variables that do not store actual data but instead store the memory address where data is located in the computer's memory. For example, if you have a number 20, and its address is 1101, the pointer will hold the address, 1101 where 20 is stored, instead of the number itself (20). Pointers are used for managing memory and linking information together.

e. **Boolean** data types represent true or false values. These are used in decision-making processes within a program, most commonly in conditional statements.

Introduction to Data Structures

Imagine your school library has different sections for different types of books such as Computing, Geography, Mathematics and Science. Each section is like a data structure, helping you to organise and find information quickly. A data structure, however, is a special way of organising, managing and storing data which makes it easier for computers to find, use, and manage data. There are common data structures that are used in computing, especially in the area of software development. Examples include lists, arrays, and trees.

Types of Data Structures

Data structures are grouped into sequential, tree-based, and graph-based and they are discussed below.

1. Sequential Data Structures

Sequential data structures store data in a linear order, where each element is connected to its previous and next element. These store data in a sequence or specific order, making it easy to access items by their position. Common examples

include **arrays** (like lists of items in a fixed order) and **linked lists** (where each item points to the next one in the sequence). Sequential structures are useful when we need to process items in order.

a. **Arrays:** An array is a collection of items stored in a specific order at nearby memory locations. All items in an array are of the same type, like a list of numbers or letters, and each item has an index to show its position. An array of numbers could look like this: [3, 6, 9, 12], where each number can be accessed by its position in the array.

b. **Linked Lists:** A linked list is a series of items called nodes, where each node holds data and a reference (or link) to the next node in the list. This linking allows for easy additions or removals of items in the sequence. An example of a linked list of numbers looks like: 3 → 6 → 9 → 12, where each number is linked to the next.

c. **Stacks**: A stack is a collection of elements that works following Last In First Out (LIFO) principle where the last item added (pushed) is the first to be removed (popped). Think of a stack like a pile of plates. When you add a plate on top, it is the first one you will take off.

Example:

i. **Stack of plates**: [plate1, plate2, plate3]

ii. Add (Push) **plate 4**: → [plate1, plate2, plate3, **plate4**]

iii. Remove (Pop) **plate4**: → [plate1, plate2, plate3]

d. **Queues:** A queue is a collection of elements that follows the First In First Out (FIFO) principle where the first item added (enqueued) is the first one removed (dequeued). A queue is like a line of students waiting to buy waakye, where the first student in line is the first to be served and be out of the queue.

Example:

i. **Queue of Student**: [Student1, Student2, Student3]

ii. Add (Enqueue) Student4: → [Student1, Student2, Student3, **Student4**]

iii. Remove (Dequeue) Student1: → [Student2, Student3, **Student4**]

Activity 4.4

Types of data structures

1. You will watch a brief presentation/video in class with photos and descriptions of the main types of data structures (Arrays, Linked Lists, Stacks, Queues, Trees, Graphs).
2. Make notes as you watch the presentation/video.
3. Identify and write down the differences and similarities of the various types of data structures shown.

4. Engage in a class discussion to discuss what you have learnt.

2. Tree-based Data Structures

Each node can have zero or more child nodes. These are used to represent data in a branching structure, similar to a family tree or organisation chart. A common example is the **binary tree**, where each item can have two children (left and right). Trees are helpful for organising data with a hierarchy, like file directories on a computer.

Components:

a. **Root**: The topmost 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:

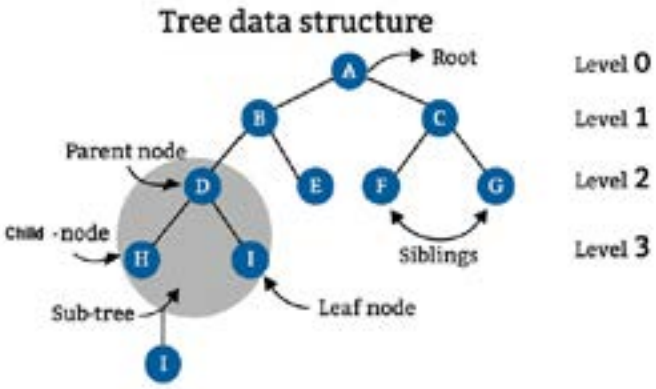


Figure 4.1: Tree Data Structure

Types of Trees

Each type of tree is purposely designed for different tasks, making it easier to store, search, or organise data effectively in computing. There are basic types of trees which are:

- **Binary Tree**: A binary tree is a tree structure in which each node has up to two children, usually referred to as the left child and the right child. Binary trees are used in searching and sorting operations.
- **Binary Search Tree (BST)**: It is a type of binary tree where each node's left child contains a smaller value, and the right child contains a larger value. This makes it efficient for search operations, as each comparison allows the search to discard half of the remaining nodes. An example of BST is shown in **Figure 4.2**.

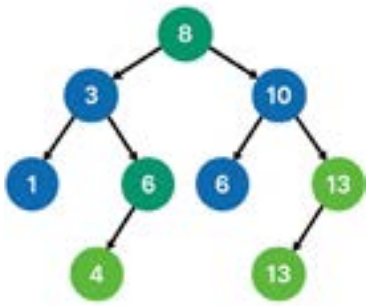


Figure 4.2: Binary Search Tree. *Image source*

3. Graph-based Data Structures

Graphs represent data as a network of connected items. Each item is called a "node," and connections between them are called "edges." This is useful for showing relationships such as social networks or maps, where locations are nodes and roads are edges. Graphs help with finding paths and understanding connections between items.

Graph: A collection of nodes (vertices) connected by edges. **Figure 4.3** is an example of a graph.

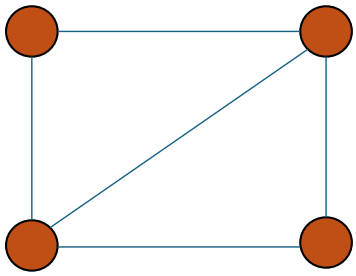


Figure 4.3: Graph

Components:

- **Vertex (Node)**: are the points or entities in a graph. Each vertex represents an item or an object in the graph, like a city on a map, a person in a social network, or a website in an internet network.

  Example: 0, 1, 2, 3, 4.

- **Edge**: Edges (or links) are the connections between vertices in a graph. An edge shows the relationship or link between two nodes.

  Example: (0-1), (0-4), (1-3), (2-3), etc.

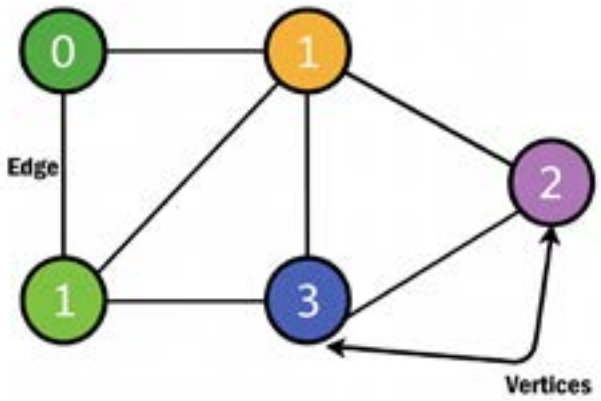


Figure 4.4: Component of Components.

Types of Graphs

There are different types of graphs. Each type serves different purposes in computing and can be applied to a variety of real-world scenarios, from mapping relationships to analysing networks. These types are:

1. **Undirected Graph**: In an undirected graph, the edges (connections between nodes) have no direction. This means you can move from one node to another in both directions. Undirected graphs are useful for representing relationships that are mutual, like friendships in a social network.
2. **Directed Graph (Digraph)**: In a directed graph, each edge has a specific direction, indicating a one-way relationship between nodes. This type of graph is useful for showing flows, like website links where one page links to another, but not necessarily back.
3. **Weighted Graph**: In this graph, each edge has a weight or cost associated with it, often representing distance, time, or any other measurable factor. Weighted graphs are useful for problems that involve finding the shortest path, like navigating through cities with different distances between them.

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

Activity 4.9

Primitive and non-primitive data structures

In groups of no more than five,

1. Using the internet research on the differences between primitive and non-primitive data structures.
2. Use flipcharts or a digital presentation to present your findings to the class for discussion and feedback.

Activity 4.10

Drawing Tree and Graph Diagrams

Individually,

1. Observe a visual diagram of a tree and a graph structure.
2. Practice drawing the visual diagrams in your notebooks.
3. Present your drawing to the class for feedback and discussion.

Extension Activity 1: Arrays

In groups of no more than five, demonstrate this activity to the class.

1. Arrange 15 exercise books on a desk in a straight line and label each position with a number from 0 to 14.
2. Explain to the class that each position number is an "index" in an array, and each book is an "element" stored in the array.
3. Show the class how to retrieve a book by calling out its index number.
4. Demonstrate how to "update" an element by swapping out a book at a specific index, showing how elements in an array can be changed while the positions remain fixed.
5. Tell the class:
   a. How quickly you were able to find a book using its index.
   b. What might be the limitations if you want to add or remove a book somewhere in the middle.

Extension Activity 2: Linked Lists

In groups of no more than five, demonstrate this activity to the class.

1. Get 6 books and 6 pieces of string to "link" them together.
2. Arrange the books in a line and tie a piece of string from each book to the next one in sequence, showing that each book points to the next in the linked list.
3. Retrieve the third book by starting from the first book and moving from one book to the next, following the "links" (string) to reach the target book.
4. Insert a new book somewhere in the middle, updating the links to "point" to the new book.
5. Remove a book from the chain and relink the surrounding books.
6. Tell the class
   a. How you were able to retrieve a specific book easily.

b. How adding or removing a book feel different from the array.

Extension Activity 3: Stacks and queues

1. In groups of no more than five, discuss
   a. how stacks are used in real life, such as the "Back" button in a web browser, which allows users to go back through their last pages visited, or in call logs, where the most recent call is shown first.
   b. real-world examples such as waiting in line at the canteen, traffic at a tollbooth, or customer service lines where the first to arrive is the first served.
2. Share your findings with the class.

Extension Activity 4: Tree-based Data Structures

In groups of no more than five, use the internet to research the following:

1. The difference between the Binary Tree and Binary Search Tree.
2. How do you think Binary Search Trees are used in real-life applications?
3. Share your findings with the class.

Extension Activity 5: Graphs

In groups of no more than five, use the internet to find out:

1. How graphs can be used to represent streets, roads, or railways.
2. How graphs can be used in recommender systems.
3. What the advantages of using a graph to represent a network of people are.
4. Five examples of real-life problems that graphs can solve and demonstrate how graphs be used to solve such problems.
5. Present your findings to the class.

HASH-BASED STRUCTURES

Hash-based structures are structures that work by using a special function to turn data into a unique number called a hash code. This code tells the computer exactly where to store or find that data within the structure. These structures are designed to make

finding, adding, or deleting data much faster. Due to this, they are used for quick data access in computing.

Hash Tables and Maps: In computing, hash-based structures such as hash tables and maps are used for organising information in a way that makes it quick to find, add, or remove data.

1. **Hash Tables:** A hash table is a data structure that stores data by using a special formula (hash function) that turns data into a unique position or "address" in a list. This helps the computer to quickly find where the data is kept. Think of a hash table as organising items on labelled shelves in your provision shops, so you can quickly find what you need.

   Components:

   - **Hash Function**: This is the formula that gives each piece of data its unique position. It converts a given key into a specific index within an array.
   - **Buckets**: The slots or spaces in the array where data is stored.
   - **Keys**: They are unique labels for each piece of data. They are also unique identifiers used to access values in the hash table. An example is a student ID number.
   - **Values**: The actual data or information stored about an entity. Example is the name of a student. The student's name is the actual data stored in the hash table.

   Example:

   - **Keys and Values**: Imagine you have a list of students, and each student has a unique ID number (key) and name (value). Let us have a look at how such information can be represented in hash table.
     - Key: Student ID
     - Value: Student Name

Table 4.1: Hash table showing Student ID and Name

Student ID	Student Name
111	Kweku
112	Dangote
113	Joyce
114	Daniel
115	Raphael
116	Mark

In table 4.1, **Daniel** has an ID **114** linked to his name and it is stored in the hash table. You can quickly find Daniel's details by looking up his ID.

Sometimes two different pieces of data might accidentally end up with the same position. This is called **Collisions**. To solve this problem, we use the method of **chaining** (linking entries at the same index) or **open addressing** (finding the next available slot).

In **chaining**, we link multiple items at the same position, like stacking papers with the same ID while in **open addressing**, if one space is taken, we look for the next free space. This helps remedy issues in hash table.

Table 4.2: Example of Chaining

Index	Values
0	[Daniel, Joyce]
1	[Dangote]
2	[Millicent, Gyampo, Eric]
3	[Thomas, Felix, Esther]
4	[Raphael, Perfect]
5	[Mark]

2. **Maps (Dictionaries):** Maps are a list of paired items where each item has a unique "name" (key) connected to a specific "thing" (value). Maps are similar to hash tables and are used to store paired information.

   An example of a map or dictionary is a Phone Book: It is similar to the hash table where it stores phone numbers and names of individuals. In a phone book, we have

   - **Key**: The phone number.
   - **Value**: The person's name.

phone_book = {

"0244123456": "Daniel",

"0200000007": "Raphael",

"0277000000": "Mark"

}

In this phone book, if you want to find Daniel's contact, you just look up his phone number.

Real-Life Applications of Data Structures

Data structures are not just for computers; they are also used in real-life systems that we interact with every day. Here are some examples:

a. **Arrays:** Like a menu of our dining hall of the school, each day of the week can be stored in an array, with meals listed as elements**.**

b. **Linked Lists:** stores tasks in a way where each task points to the next one. This makes it easy to add, remove, or rearrange tasks throughout the day.

c. **Stacks:** Web browsers use a stack for backtracking. Each webpage visited is added to the stack, and pressing "back" pops the last page off, thus going back to previous pages.

d. **Queues:** Banks such as GCB, ADB, CBG, Ecobank, etc and service centres such MTN, Vodafone and AT offices in Ghana use queues to manage customers, ensuring they are served in the order they arrived.

e. **Trees:** In a school, the organizational chart is a tree where the headmaster is at the top, with branches extending to assistant heads and other subordinates.

f. **Graphs:** A **graph** can model connections, such as bus or subway routes in a city. Each station or stop is a node, and the road connecting them is an edge.

g. **Hash Tables:** In a school, each student's ID number could be used to quickly find their information.

h. **Maps (dictionaries):** A contact list on a phone is a map where each name (key) is linked to a phone number (value).

Activity 4.13

Real Life Application of Data Structures

Organise yourselves into groups of no more than five,

1. In your groups, briefly brainstorm and discuss how different data structures (Arrays, Linked Lists, Stacks, Queues, Trees, Graphs, Hash Tables) are used in real-life. Engage in an initial discussion with the entire class to share your ideas.
2. Now, use the internet and research on how arrays, linked lists, stacks, queues, trees, graphs, and hash tables are used in real-life.
3. Note down at least five real-life applications of these data structures and explain in your own words how these data structures are used.
4. Present your findings to the class for feedback and discussion
5. Individually, reflect on what you learnt and write a short note on any new insights gained from the research and presentations.

EXTENDED READING

- Click *here* or https://www.youtube.com/watch?v=knV86FlSXJ8 on to watch video on hash tables.
- Click *here* or *https://www.youtube.com/watch?v=5ltBWq3cUwM* to watch a video on data types in computing.
- Click *here* or *https://www.youtube.com/watch?v=cQWr9DFE1ww* to watch video on data structures in computer science

Review Questions

1. What does RAM stand for?
2. Identify one type of RAM.
3. What is a tree in data structures, and how is it used in organizing information?
4. Assuming you are organizing a school's timetable. Which data structure would be most useful, and why?
5. How is a linked list different from an array, and give a real-life example of a linked list?
6. Describe a queue and give an example of its use in everyday life.
7. How can a queue be used in customer service, and what is the advantage of this structure?
8. Let say you are designing a recipe book app that lets users browse and search for recipes by ingredients. What data structure(s) would best allow for quick search and retrieval based on ingredient types?
9. For an online store's self-checkout system like Jumia, Melcom, where items are scanned and placed in a virtual cart, which data structure would you choose to represent the cart, and why?
10. Kofi is chosen to design a scheduling system for inter school's sports tournament to be held at Tumu where teams advance in a knockout format, which data structure would you advise. Kofi to use to manage the rounds?
11. How would you optimise a university's course registration system to handle high traffic on the first day of enrolment? What data structures would you choose to prevent server overload and manage student requests effectively?
12. For a disaster response app, you need to track the location of rescue resources (like ambulances and shelters) and update their status in real time. How would you use data structures to optimise resource allocation?
13. How would you use data structures to create an intelligent spell-checking feature that can handle common spelling errors, suggest corrections, and remember user preferences for future use?

SECTION

5

UTILISING PROGRAMMING LANGUAGE

COMPUTATIONAL THINKING (PROGRAMMING LOGIC)

App Development

INTRODUCTION

In this section, you will learn how to create a simple calculator program and utilise simulation tools in machine learning to train computers to perform specific actions or sets of actions. You will get hands-on experience with Python and discover how to use essential programming concepts like variables, decision-making, loops and functions.

By the end of this section, you will have the skills to write your own Python programs, build a calculator that can perform basic arithmetic tasks and perform machine learning tasks.

Key Ideas

- **A conditional statement** is a way for a computer to make decisions.
- **A variable** is like a storage box in a computer where you can keep different types of information like numbers or words.
- **Machine Learning** is when computers learn from data without being told exactly what to do.

BASIC PROGRAMMING CONCEPTS

Variables

Imagine a box with a label on it. This box can hold anything you put inside – like a toy, a book, or snacks. In programming, a variable is just like that box! It is a container where you can store information, such as numbers, words, or other data.

However, there are some conditions to the naming of variables.

1. **Meaningful Names:**
   Pick names that explain what the variable means. This makes your code easier to read and understand. For example:

a. `**age**` for a person's age
b. `**total price**` for the cost of items
c. `**student name**` for a learner's name

2. **Case Sensitivity**
   In Python, variable names are case-sensitive. This means: `**age**`, `**Age**` and `**AGE**` are treated as different variables.

   Note: This may vary in other programming languages.

3. **Start with a Letter or Underscore**
   Always begin variable names with: A letter (like `**a-z**` or `**A-Z**`) or an underscore (`_`). You cannot start with a number!

4. **No Spaces or Special Characters**
   Variable names cannot have spaces or symbols like `**@**`, `**#**`, `**$**`, etc, except for underscores (`_`).

5. **Use Underscores for Readability**
   When naming variables with multiple words, use underscores to separate the words and make them clear, for example: `**student age**`, `**total price**`. This naming style is called **snake case.**

6. **Avoid Reserved Keywords**
   Stay away from names that are reserved by the programming language, such as: `**if**`, `**else**`, `**while**`, `**for**`, etc.

   These keywords have special meanings and cannot be used as variable names.

   Following these variable tips will keep your code clean and easy to understand!

Activity 5.1

Writing Variables

1. Click to open this URL (*https://www.programiz.com/python-programming/online-compiler/*)
2. Copy and paste the codes below into main.py and **RUN** it
   FirstName = "Raphael"
   **FirstName = "Senyo"*
   age1 = 17
   3age = 20
   School Name= "Zavi SHTS"
   _Hometown= "Dambai"
   In = football
3. Study the output error and use your knowledge of variables to correct it.

4. Share your result with your peers in class.

Data Types

Data types tell us the kind of information a variable can hold. For example, is it a number, a word, or something more complex? Understanding data types helps us write accurate and reliable code.

Types of Data

1. **Primitive Data Types**: These are basic and simple types like:
   a. Numbers (Integer and float): e.g., 5, 3.14
   b. Strings (e.g., "hello", "Ghana")
   c. Boolean (True or False): e.g., True, False
2. **Complex Data Types**: These are more advanced types like:
   a. Arrays (e.g., [1, 2, 3])
   b. Lists (e.g., "Daniel", "Raphael", "Mark", "Millicent")
   c. Dictionaries (e.g., {name: "Kwame", age: 16})
3. **Comments**

   Comments are little notes that you leave in your code to explain what is happening. They do not affect how the program runs because the computer ignores them. They are just there for humans!

   We use comments to:

   a. explain what a section of code does.
   b. remind yourself why you wrote the code a certain way.
   c. help others (or your future self) understand your code quickly.

   How to Add Comments:

   In Python, there are two ways of writing comments,

   1. A single-line comment (makes use of the # symbol for a comment on a single line)

      # This calculates the total price of items.

      total price = price * quantity

   2. A multi-line comment (makes use of opening and closing quotes typed three times).

      ''' The following code is

      meant to get data about oneself'''

      FirstName = "Raphael"

Middle_Name = “Senyo”

MyAge1 = 17

MyName: “Zavi SHTS”

_Hometown: “Dambai”

Activity 5.2

Commenting

Individually or in pairs:

1. Copy the code below into a Python editor (e.g., Thonny, PyCharm, or any online Python compiler).
2. Run the code and observe the output.
3. Read the comments in the code to understand what each part does.
4. Write additional code to calculate a discounted cost of 10%.
5. Use a multi-line comment to explain how you did the calculation. Discuss your results with the class.
6. price = 50 # Ghana Cedis
   quantity = 3 # Number of items bought
   # Calculate the total cost
   total_cost = price * quantity # Multiplying price by quantity
   # Print the result
   print(“The total cost is:”, total_cost)

Functions in Programming

A function is a block of code that performs a specific task. It helps us organise work and makes programming easier to manage. We use functions for the following reasons:

1. **Reusability**: Write code once, then use it again and again!
2. **Organisation**: Break bigger problems into smaller and easier parts.
3. **Readability**: Code will be easier to read, understand and fix.
4. **Avoid Repetition**: We use functions to handle tasks that repeat, so we do not have to write the same code over and over.

Functions have two structures

1. **Function Definition**: This is where you create the function.
2. **Function Call**: This is where you use the function in your program.

Let us now look at how to define a function in python and create some.

In Python, we write functions like this

```
def function_name (parameters):
# Write your code here
return value
```

Note: parameters are the values that are passed into the function so that it can do what it has been programmed to do. Not all functions need to have parameters.

Activity 5.3

Creating Functions 1

1. Copy and paste this code into a Python editor and run it.

```
def sum_of_numbers (a,b):    #defining the function
result = a + b
print (result)
sum_of_numbers (3,4)   # Recalling the function "sum_of_numbers"
"""anytime you recall a function, you only have to specify the parameters like how we          did for 3 and 4."""
```

2. Recall the function for -4, 4
   You output should be "0".

Activity 5.4

Creating Functions 2

1. Copy and paste this code into a Python editor and run it

```
def subtraction_of_numbers (a,b):    #defining the function
result = a - b
print (result)
subtraction_of_numbers (3,4)   # Recalling the function "subtraction_of_numbers"
"""anytime you recall a function you only have to specify the parameters like how we did for 3 and 4"""
```

2. Write your reasons for why this program does not run?
3. Debug this code by comparing it to **Activity 5.3**.
4. Recall the function for 0 and 14.

Activity 5.5

Function Hurdle Race Game

The goal of the game is to guide Reeborg to the winning flag.

1. Click *here* or (*https://reeborg.ca/reeborg.html?lang=en&mode=python&menu=worlds%2Fmenus%2Freeborg_intro_en.json&name=Hurdle%201&url=worlds%2Ftutorial_en%2Fhurdle1.json*) to open the game.
2. Click *here* or (*https://youtu.be/rfckMNNoX5g*) to view the instructional information in a new browser tab.
3. A possible solution to this activity is embedded in this link:

   https://youtu.be/YDUbTwHPhmc or *here*.

Arithmetic operations 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

Activity 5.6

Applying Arithmetic Operation in Python

Launch a Python complier and complete the following:

1. Write a Python program to calculate the total cost of buying:
   3 loaves of bread costing GHS 10 each.

2 bottles of water costing GHS 4 each.

Hint: Use (+) to sum the costs and display the results.

2. Write a Python program to calculate the remaining credit on a prepaid SIM card.

   Starting balance: GHS 50.

   Call costs: GHS 15.

   Hint: Use subtraction **(-)** and display the results.
3. Write a Python program to find the total cost of buying 8 mangoes, each costing GHS

   Hint: Use multiplication (*).
4. A box of chocolates contains 24 pieces. Write a Python program to calculate how many chocolates each of 6 friends will get if shared equally.

   Hint: Use division (/).
5. Using the modulus (%), write a Python program to find the remainder of 17, when divided by 4.
6. Using the Python language, write a program that asks for four numbers, calculates the first two and multiplies the last 2.

Logical Operation

Logical operators are tools in programming that help make decisions based on conditions. They work with true or false values to combine or compare them. For example:

1. AND: Checks if both conditions are true.
2. OR: Checks if at least one condition is true.
3. NOT: Reverses the value (true becomes false, and false becomes true).

These operators are used in situations where we need to check multiple conditions like in if-else statements.

Comparison Operators

Comparison operators are symbols used in programming to compare two values. They help us make decisions in our code by checking relationships between numbers or other data. Let us look at each one:

1. **Equal to (==)**:

   This operator checks if two values are the same. For example: Is 10 the same as 10? Yes, so it returns True.

   Example in Python:

   ```
   is_equal = (10 == 10)  # is_equal is True.
   ```

2. **Not equal to (!=)**:
   This checks if two values are different. For example: Is 8 different from 4? Yes, so it returns True.
   Example in Python:
   is_not_equal = (8 != 4) # is_not_equal is True.

3. **Greater than (>)**:
   This checks if one value is bigger than another.
   For example: Is 13 greater than 2? Yes, so it returns True.
   Example in Python:
   is_greater = (13 > 2) # is_greater is True

4. **Less than (<)**:
   This checks if one value is smaller than another.
   For example: Is -3 less than 0? Yes, so it returns True.
   Example in Python:
   is_less = (-3 < 0) # is_less is True

5. **Greater than or equal to (>=)**:
   This checks if a value is bigger or the same as another.
   For example: Is 7 greater than or equal to 5? Yes, so it returns True.
   Example in Python:
   is_greater_or_equal = (7 >= 5) # is_greater_or_equal is True

6. **Less than or equal to (<=)**:
   This checks if a value is smaller or the same as another.
   For example: Is 4 less than or equal to 6? Yes, so it returns True.
   Example in Python:
   is_less_or_equal = (4 <= 6) # is_less_or_equal is True

Activity 5.7

Comparison Operators

1. Determine whether the following comparisons are true or false:
   a. 8 == 8
   b. 10! = 5
   c. 12 > 15
   d. 7 < 9
   e. 6 >= 6
   f. 4 <= 3

Control Structures

Conditional Statements

Conditional statements in programming allow a program to make decisions and follow different paths, depending on whether certain conditions are true or false. Think of conditional statements like choosing which road to take at a junction based on the traffic light's colour.

Types of Conditional Statements

1. **if Statement**

   The if statement is used to run a specific block of code only if a condition is true.

   Example:

   Height = 45

   if Height > 30:

   print("He is very tall")

2. **if-else Statement**

   The if-else statement is used when there are two possible outcomes: one for when the condition is true and another for when it is false.

   Example:

   Height = 45

   if Height > 30:

   print("He is tall")

   else:

   print("He is not tall")

3. **if-elif-else Statement**

   The if-elif-else statement is used when there are multiple conditions to check. The program checks the conditions one by one until it finds one that is true.

   Example:

   if Height > 30:

   print("He is tall")

   elif Height > 20:

   print("He has a normal height")

   else:

   print("He is short")

Activity 5.8

If statements

Fix the following code to make it work:

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

Activity 5.9

Logical Operations and if statements

1. Create a Python program that checks if two conditions are true to unlock a "treasure chest".

   Write a program where both conditions must be met:

   Condition 1: age >= 18.

   Condition 2: has_ticket == True.

   If both are true, print: "Access Granted: Treasure Unlocked!"

   If not, print: "Access Denied: Try Again."

2. Write a Python program to display: "Congratulations! You qualify for the scholarship." if either condition below is true, and "Sorry, you do not qualify." if both are false.

   The student qualifies if they either have:

   Condition 1: An average score above 80, OR

   Condition 2: Excellent sports performance (sports_award == True).

3. Complete the following code to toggle the state of a light, using the NOT operator to invert the current state of the light (is_light_on).

   ```
   is_light_on = False
   is_light_on =
   if is_light_on:
   print("The light is now ON.")
   else: print("The light is now OFF.")
   ```

4. Study this program and write your reflection of the code:

   ```
   a = float(input("Enter the first number: "))  # a is for the first input
   b = float(input("Enter the second number: "))  # b is for the second input
   if a == b:  # Checking if a and b are equal
     print(f"{a} is equal to {b}")
   elif a > b:  # Checking if a is greater than b
   ```

```
    print(f"{a} is greater than {b}")
else:
print(f"{a} is less than {b}") # Checking if a is less than b
```

Based on your understanding of item 4, write a Python program to compare.

a. The heights of three students and find the tallest.

b. Compare three prices and find which one is the cheapest.

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.

Loops and Recursion

1. **For Loop:** A for loop is a way to make a computer repeat a task a specific number of times. It is very useful when you already know how many times the task needs to be done.

   Think of it like counting steps when climbing a staircase. If the staircase has 10 steps, you know you need to step 10 times, and you can use a for loop to count each step.

   Example in Python:

   ```
   for step in range(1, 4):
   print("Step", step)
   ```

   This code prints:

   Step 1

   Step 2

   Step 3

   Note: The range (1, 4) gives the numbers 1, 2 and 3. Python stops one number before 4, so the loop does not include 4.

   While Loop

   A while loop keeps repeating a task as long as a condition is true. It is like saying, "Keep adding water until the bucket is full." You do not know how many times it will run; it stops when the condition becomes false.

   Example:

   ```
   count = 1
   while count <= 5:
   print(count)
   count += 1 # Add 1 to count each time
   ```

Activity 5.10

Loops

1. Write a Python program that prints numbers from 1 to 20, using a for loop.
2. Write a Python program to display the multiplication table of 5, using a for loop.
3. Run the below code in a complier for Python.
   a. Enter wrong password three times.
   b. Enter the correct password.
   c. Analysis the code carefully and write your reflection on why it works like that.

```
password = "1234"   # correct password
guess = ""
while guess != password:
guess = input("Enter the password: ")
  if guess == password:
     print("Access granted!")
  else:
     print("Incorrect password. Try again.")
```

2. Recursion

Recursion is a technique in programming where a function calls itself to solve a problem. It is useful when a big task can be broken down into smaller and similar tasks. The function keeps calling itself until it reaches a condition that stops it, it's called the base case.

For instance, if you want to calculate the factorial of a number (e.g., $6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1$), recursion can do it by repeatedly multiplying the number by the factorial of the number just below it.

```
def factorial(n):
 if n == 1:  # Base case
      return 1
     else:   # Recursive call
      return n * factorial(n - 1)
 print(factorial(6))  # Output: 720
```

Activity 5.11

Recursion

1. Debug this recursion code:

```
def countdown(n):
  if n == 0:  # Base case
    print("Blast off!")
  else:
    print(n)
    countdown(n + 1)
```

2. Write a Python function to calculate the sum of numbers from 1 to n using recursion.

Creating a Simple Calculator

In creating a simple calculator, there are some key steps we need to follow to make the calculator user friendly. The steps are:

1. **Prompt for User Input**: In Python, the input() function allows a program to ask the user for information while it is running. The user can type in their response and the program can then use that information. For example:

   Num1 = input("Enter your first number: ")

   When a user enters input in Python using the **input()** function, it is always interpreted as a string, even if the input looks like a number. To use the input as a number for calculations, you need to convert it using **int()** (for whole numbers) or **float()** (for decimal numbers).

   Using **float()** is generally preferable because it handles both integers and decimals. For example, if a user enters the number 10, float() will still work and treat it as 10.0, while also accommodating inputs like 10.5. This flexibility ensures the program can handle a wider range of numeric inputs.

   You can convert the string to a float by writing the word float directly in front of the input() function, as the below code:

   num1 = float(input("Enter your first number: "))

2. **Enter the Operator**: In programming, we can ask the user to input an arithmetic operator which we studied earlier (+, -,*,/,**).

   When the user types one of these symbols, the program will store it in the assigned **variable** (let us call ours 'operator'). This variable holds the user's choice so the program can use it later to perform calculations. For example:

   num1 = float(input("Enter your first number: "))

   operator = input("Enter any of these operators (+, -,*,/,**)")

3. **Prompt for the Second Number**: Repeat the code for the first number but this time change the variable.

```
num1 = float(input("Enter your first number: "))    # First number
operator = input("Enter any of these operators (+, -,*,/,**)")
num2 = float(input("Enter your first number: "))    # Second number
```

4. **Performing the calculation**: we can use conditional statements like if, elif, and else to make decisions and perform different tasks based on the input provided by the user.

5. **Displaying the Result**: Use the print to display the answer.

Note that when writing the conditional statements, you have to write the operation symbols in quotes because they are stored as strings.

```
num1 = float(input("Enter your first number: "))    # First number
operator = input("Enter any of this operators (+, -,*,/,**): ")
num2 = float(input("Enter your second number: "))    # Second number
'''Using the conditional statement to perform the calculation'''
if operator == '+' :
result = num1 + num2
print("Answer: ", result)
elif operator == '-':
result = num1 - num2
print("Answer: ", result)
elif operator == '*' :
result = num1 * num2
print("Answer: ", result)
elif operator == '**' :
result = num1 ** num2
print("Answer: ", result)
elif operator == '/' :
if num2 == 0:
print('Division by zero is not possible')
  else:
result = num1 / num2
print("Answer: ", result)
else:
print("Ooops, I think there is a mistake with the operator you entered")
```

Activity 5.12

Creating a simple calculator

In groups of no more than five,

1. Discuss the features you would include in a calculator.
2. Create a flow chart outlining the logic and flow of a calculator programme,
3. Study the code above and write a Python program that can calculate 6 numbers.
4. Present this to the class for feedback and discussion.

UNDERSTANDING MACHINE LEARNING CONCEPTS

Machine Learning (ML) is a part of Artificial Intelligence (AI) that teaches computers to learn and improve by using data, without being directly told what to do. Think of it like how a student improves in maths; the more they practise solving problems, the better they become. Similarly, computers "learn" by practising with large amounts of data.

Here are some examples of how Machine Learning is used in everyday life:

1. **Social Media Feeds**: Apps like Facebook or Instagram show you posts based on your interests. They "learn" what you like by studying the posts you interact with.

Figure 5.1: Social Media Feed Machine Learning Process

2. **Online Shopping Recommendations**: Websites like Jumia suggest products you might want to buy, based on what you have searched for or purchased before.

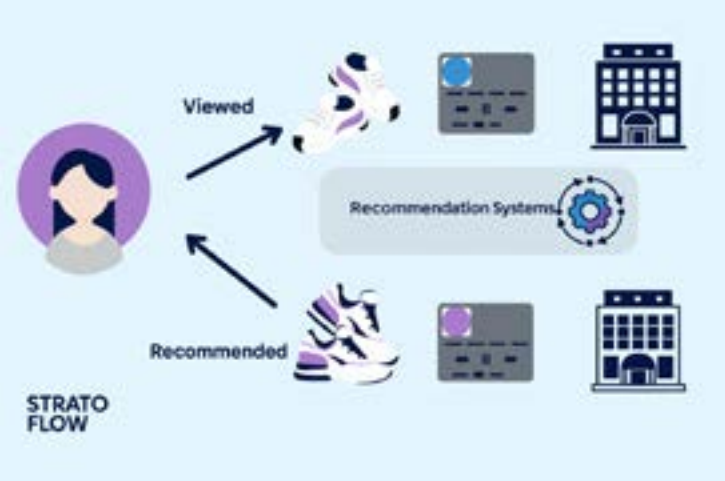


Figure 5.2: Online

3. **Voice Assistants**: Tools like Siri, Ella, Google Assistant or Gemini can understand your voice commands and get better at recognising your speech over time.

Figure 5.3: Gemini Voice Assistance

4. **Spam Email Filters**: Machine Learning helps email systems detect and block unwanted spam messages automatically.

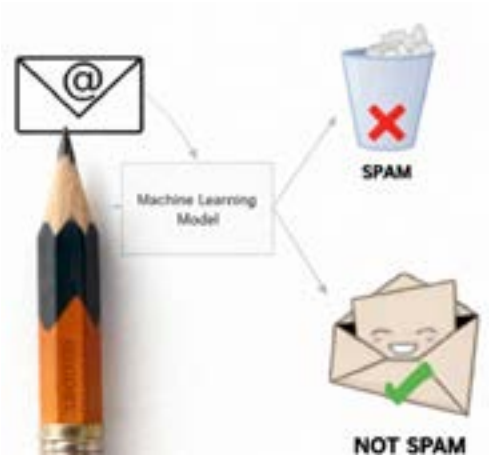


Figure 5.4: Diagram of Email Spam Filter

5. **Predicting the Weather**: Machine Learning analyses weather data to make accurate forecasts.

Figure 5.5: ML Diagram for Weather Predictions

Activity 5.13

Machine Learning Concepts

1. In groups of no more than 5, list and explain four additional examples of how Machine Learning is used in everyday life.
2. Share your findings with the class.

Types of Machine Learning

There are three main types of Machine Learning (ML), each are based on how machines learn from the data inputted:

1. **Supervised Learning:** In supervised learning, the machine learns from **labelled data**, where the correct answers (outputs) are provided. Here is an example: Teaching a computer to recognise fruits by giving it pictures labelled "apple" or "orange."

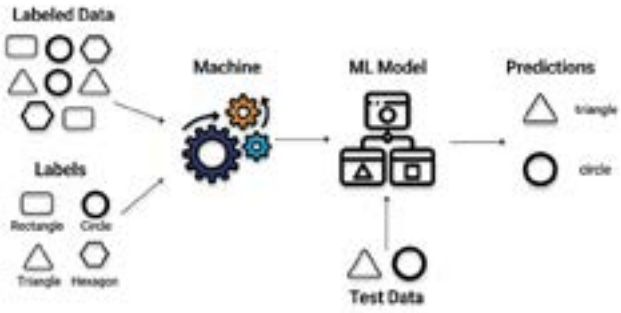


Figure 5.6: Supervised ML Diagram

Examples

- **Image Classification** – for example, identifying different types of animals
- **Predicting House Prices** – using labels like size and location to estimate the price of a house

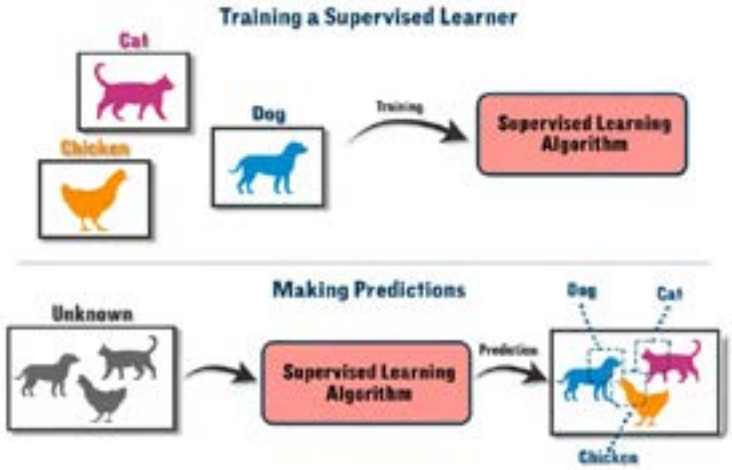


Figure 5.7: Types of Supervised Learning

In **Figure 5.7**, the data is labelled before passing it through the model.

2. **Unsupervised Learning:** The machine is given unlabelled data and has to find patterns or groupings on its own. Here is an example: Grouping customers with similar shopping habits without knowing their preferences beforehand. The goal is to identify relationships within the data without knowing anything about it beforehand. This is the opposite to supervised learning, where the model is trained on a labelled dataset.

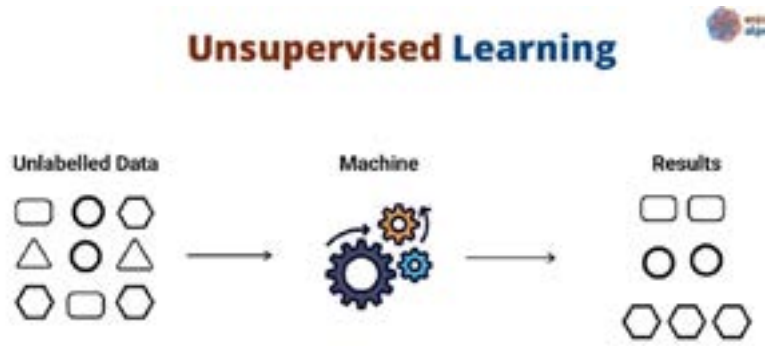


Figure 5.8: Unsupervised ML Diagram

Clustering is a common task in unsupervised learning. Clustering is the process of diving data into groups (clusters) based on their similarities. Each group contains data points that are like each other. The data points are different across different groups. Clustering works by analysing the data points to calculate how similar or different they are from each other; this is usually measured using distance. The data points that are close together are considered similar and are then grouped into the same cluster.

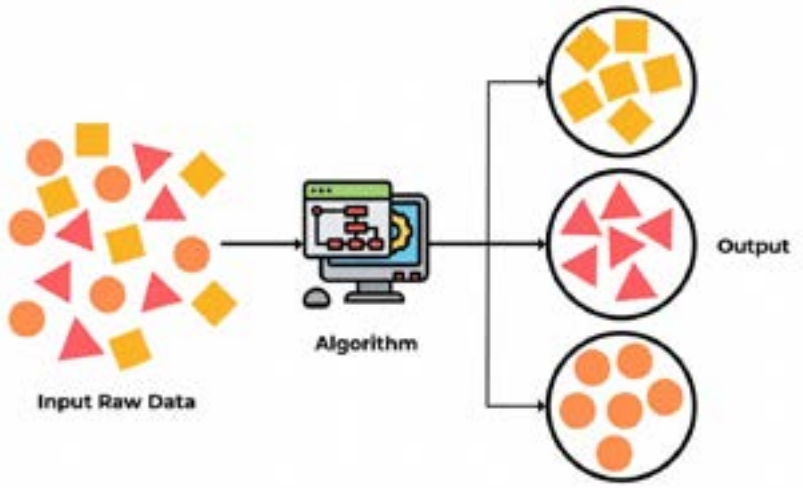


Figure 5.9: Clustering

Applications of Clustering

Clustering can have a wide range of uses, including:

- **Customer Segmentation –** business can group customers together based on certain factors, such as what they purchase, where they live or their preferences. This can then be used to target marketing campaigns.
- **Image Segmentation –** clustering can be used to identify different objects of regions within an image.
- **Document Clustering –** large collections of documents can be organised into clusters based on their content, making it easier and quicker to search for relevant information.

- **Anomaly Detection** – businesses could use clustering to identify outliers that don't belong to any cluster, this helps to detect unusual data points that could be errors or fraudulent activities.

Common Clustering Algorithms

- **K-mean Clustering –** dividing data into a predefined number of clusters.
- **Hierarchical Clustering –** creating a hierarchy of clusters, starting with individual data points and merging them based on similarities.

3. **Reinforcement Learning:** In reinforcement learning, the machine learns by trial and error, receiving rewards for correct actions and penalties for wrong ones; an example is: a robot learning to navigate a maze by getting points for moving closer to the exit.

Figure 5.10: Reinforcement ML Diagram

a. **Agent –** the learner or decision maker
b. **Environment –** the external system that the agent interacts with
c. **Action –** what the agent can do
d. **Observation –** the current state of the agent
e. **Reward –** the feedback from the environment

In the above example, the dog is the agent and is interacting with the human as the environment. The agent has two actions; it can either catch the stick or not. Depending on the action and the observation, the reward would be either a treat/ praise if it catches the stick or nothing if it fails.

Real-World Examples of Machine Learning

Training self-driving cars to navigate roads, obey traffic laws and react to dynamic elements such as traffic lights, pedestrians and other vehicles on the road

Training robots and drones to navigate and perform tasks in dynamic and uncertain environments

Data Collection and Preparation

To teach a machine learning model, we must give it high-quality data so it can understand patterns and make predictions. The quality and amount of data are very important. Good data helps the model perform well, while poor data leads to bad results. For example, if we want to train a model to recognise objects in pictures, we need to collect many images that represent real-life objects.

The basic steps to collect and prepare data for a machine learning model include:

1. **Define the Problem:** Decide what you want the model to do. For example, do you want it to recognise animals in pictures or predict the weather?
2. **Collect Data**: Gather enough examples of the information you want the model to learn. For an image-based task, this could involve downloading pictures or taking your own.
3. **Clean the Data**: Remove errors or incorrect information. For instance, if your data contains blurry images, irrelevant images, or duplicates, you should get rid of them.
4. **Label the Data:** Organise the data by adding labels that describe it. For example, if your images show cats and dogs, you can label them as "cat" or "dog."
5. **Split the Data**: Divide the data into two main parts:
   a. Training Data: This is used to teach the model.
   b. Testing Data: This is used to check if the model has learned correctly.
6. **Normalise the Data**: Prepare the data in a standard way to help the model learn more effectively. For images, this might mean resizing them to the same size.

What are Machine Learning Libraries?

ML Libraries are pre-written code snippets that are stored in re-usable collections. They aim to help developers to implement machine learning algorithms quickly and efficiently. The collections can help simplify the process of developing, training and deploying machine learning models. Below are some examples of some basic machine learning libraries:

NumPy

This is a tool that helps perform computations (for example, mathematical equations). It is a Python library which is designed to work with numbers and arrays. NumPy is good at performing various mathematical operations quickly and efficiently, making the process straightforward and fast.

Example uses: NumPy could be used to add up all the numbers held within a large list or carry out complex calculations.

Pandas

This is a tool very similar to NumPy, however it is an upgraded version which has been tailored for handling structured data, like the tables in an Excel spreadsheet. Pandas is designed to help organise and analyse data arranged in rows and columns. It is built upon NumPy and adds an extra layer of functionality, which can be used in data analysis activities,

Example uses: Pandas could be used to clean large datasets, identifying and dealing with missing values, to ensure that the data is accurate.

Scikit-Learn

This is a tool which is often referred to as Sklearn. It is an easy-to-use open- source Python library, which provides simple and efficient tools for data mining and data analysis. The Sklearn library contains a lot of efficient tools for machine learning and statistical modelling, including classification, regression, clustering and dimensionality reduction.

Example use: Sklearn could be used when providing users with recommendations on platforms such as Netflix or Spotify as it can be used to analyse trends in usage.

Activity 5.14

Exploring Machine Learning

1. Open this google drive link and download the folders
   https://drive.google.com/drive/folders/1avSlyyLdPmpYTxsvFL5s_6FktPIp2aj8?usp=drive_link
2. Launch this web application in a new tab
   https://machinelearningforkids.co.uk/#!/welcome
3. Click on “get started”
4. Click on “try now” adjacent “try without registration”
5. Click on “add a project”
6. Enter the project name as “Books & Pens Model”
7. Under project type, select “recognising images”
8. Under storage, select “in your web browser”
9. Then click on create
10. Click on Books & Pens Model
11. Click on “Train from the displayed options”
12. Click on add new label
13. Type “Book” into the text field and press Enter/ Click create
14. Click on “file” at the bottom of the displayed box

15. Navigate to the download folders, select all the images in books folder and click on "open" for the images to upload into the label "book"
16. Repeat step 12 to 15 for pen and eraser
17. Click on "back to project" at the left top corner
18. Click "Learn & Test"
19. Then "Train new machine learning model"
20. Test the model by using the webcam option (show objectives like books, pens or erasers to the webcam) and see the response of your model.
21. Or test the model by using the draw option (make drawings that looks like a pen, eraser or a book on your screen) and see the response of **your model.**

Key components of Machine Learning Model

1. **Data:** This is the information we give the computer to learn from. It can be:
   a. Structured data like tables and charts (e.g., a student's scores in maths and science.
   b. Unstructured data like pictures, text, or audio (e.g., like the images we used in **Activity 5.14**.
2. **Features:** Features are the important details or characteristics of the data used to make predictions. For instance:
   a. In our activity we used labels (book, pen and eraser).
   b. To make predictions, we improve these features through feature engineering, which is selecting and refining the data.
3. **Algorithm:** This is the "recipe" or set of rules the computer follows to learn. Examples include:
   a. Linear regression (predicting continuous numbers like a student's average score).
   b. Decision trees (making yes/no decisions).
   c. Clustering algorithms (grouping students based on similar performance).
4. **Training:** Training is teaching the computer using data.

   We divide the data into two parts: one for training (teaching the model) and one for validation (testing how well it learned).
5. **Evaluation:** After training, we check if the model works well. We measure performance using:
   a. Accuracy (how many predictions were correct).
   b. Precision (how often predictions were right when they were made).
   c. Recall (how many true results were identified).

6. **Prediction:** Once the model is trained and tested, it can make predictions with new data. For example: Predicting a student's future grades based on their past scores.

Activity 5.15

Classification of Data

1. Imagine you are creating a machine learning model to predict a student's success in Ziavi SHTS. Below are some examples of data:
   a. A table showing student names, scores and attendance percentages.
   b. A group photo of students in their classroom.
   c. An audio clip of a student giving a presentation.
   d. A paragraph describing a student's achievements in school.
2. Use the information in item (1) to group the data into structured data and unstructured data.
   Hint: Structured data is organised in rows and columns (like tables) unstructured data does not fit neatly into a table (like pictures or audio).
3. Write a short explanation of why each piece of data fits into its group.
4. Share your work with peers in class.

Extension Activity 1: Imagine a Ghana with Machine Learning

Write a short story or a paragraph about how ML could change Ghana in the next 10 years in the fields of healthcare, farming and education.

IMPLEMENTING MACHINE LEARNING AND ETHICAL CONSIDERATION

Activity 5.16

Practising Machine Learning

In this activity, we will use Google's Teachable Machine tool to build a model that can tell the difference between a dog and a human. It is simple because you do not need to write code—just use your webcam to train the computer! Follow these steps to create your model:

Steps to Train the Model:

1. Open a browser then click *here* or visit teachablemachine.withgoogle.com.
2. Select the Image Project option, as we are working with pictures.
3. Create two categories (or classes) to train your model: one for "dog" and another for "human".
4. Use your webcam or upload pictures.
5. For the "dog" class, take or upload images of dogs.
6. For the "human" class, take or upload images of humans.
7. Ensure you collect enough samples (about 15–20 images per class).
8. Click the Train Model button. The tool will process the images and create a model that can classify between books and pens.
9. Use your webcam or upload images to see if the model correctly identifies the object as a "dog" or a "human".
10. If the model makes mistakes, add more training data and retrain it.

Ethical Considerations in Machine Learning Models

Ethics are the rules or principles that help us decide what is right or wrong. They guide how we act to make sure we treat others fairly with respect and do what is just.

When it comes to machine learning, ethics means using these systems in ways that are responsible and do not cause harm to people or society. For instance, it would be unethical to use a machine learning model that spreads false information or unfairly discriminates against a group of people.

We therefore, need to cautiously think about the impact of the models we create and use so as to ensure they are fair, safe and helpful to everyone.

Let us look at some unethical considerations and how to fix them.

Bias in Training Data

Imagine teaching a robot how to make decisions but the examples you give it are unfair or have mistakes. For example, if the training data mostly includes people from cities and not villages, the robot might not understand the needs of people in villages. This is called bias.

Fixing Bias

To fix this, use information from different places, check for mistakes, and make sure everyone is included equally in the data.

Fairness in Machine Learning

Fairness means treating everyone equally no matter their background, gender or where they come from. A model should give fair results to everyone.

Fixing Fairness Issues

We can check the model for fairness problems, design it to avoid favouring any group and involve people from different backgrounds in its creation.

Privacy in Machine Learning

Privacy is about protecting people's personal information like health or school records, so that it does not get into the wrong hands. For example, a model using student health data should ensure no one else can see that information without permission.

Fixing Privacy Issues

To protect privacy, hide people's names and personal details (anonymise the data), only collect information that is necessary and use methods to keep it secure.

Accountability in Machine Learning

Accountability means the people who make a machine learning model must take responsibility for its mistakes. For example, if the model provides wrong information, the developers must find out why and fix it.

Fixing Accountability Issues

We need to test the models carefully; follow rules and make it easy for people to report any problems that they find.

There are some potential societal impacts of machine learning models. These are:

1. **Positive Impacts**
   a. Models can help with accurate decisions such as predicting the weather or diagnosing illnesses.
   b. Machines can handle boring tasks quickly and correctly so humans can focus on more challenging problems.
2. **Negative Impacts**
   a. If not designed properly, models might make life harder for certain groups like giving unfair treatment to disadvantaged people.
   b. Using too much personal data can lead to privacy problems such as data theft.
   c. Complex models can be hard to understand, making it tricky to trust their results.

Activity 5.17

Ethical Considerations

Individually, tick the boxes to identify the correct ethical considerations

Share your responses with your peers.

S/N	Ethical Consideration	Tick the appropriate box			
		Bias in Training Data	Fairness in Machine Learning	Privacy in Machine Learning	Accountability in Machine Learning
1	Anonymise data to hide personal details.				
2	Use data from only one group of people.				
3	Use data from multiple sources.				
4	Use checklist to identify and fix problems.				
5	Collect as much data as possible, even if it is not necessary.				
6	Test the model to ensure it gives reasonable results to all groups.				
7	Ignore unfair patterns in the data.				
8	Only use data from past trends, even if they are biased.				
9	Allow the model to operate without following any rules.				

S/N	Ethical Consideration	Tick the appropriate box			
		Bias in Training Data	Fairness in Machine Learning	Privacy in Machine Learning	Accountability in Machine Learning
10	Involve people from different backgrounds in developing the model.				
11	Test models thoroughly before releasing them.				
12	Check the model to see if it treats everyone equally.				
13	Check for errors or missing data in the training dataset.				
14	Share sensitive information freely to improve accuracy.				
15	Make it easy for people to report issues with the model.				
16	Involve a team with diverse perspectives to develop the model.				
17	Take responsibility for any harm caused by the model.				
18	Design the model to favour one group over others.				
19	Ensure the data includes all groups equally.				

S/N	Ethical Consideration	Tick the appropriate box			
		Bias in Training Data	Fairness in Machine Learning	Privacy in Machine Learning	Accountability in Machine Learning
20	Ignore errors in the training data.				
21	Developers should fix any problems their models create.				
22	Keep people's personal information confidential and secure.				
23	Ignore feedback from affected groups about fairness issues.				
24	Only collect the information needed for the model's purpose.				

Activity 5.18

In groups of no more than five, complete the following:

1. Discuss the key components of building a simple machine learning model, using the Iris flower dataset
2. Chose a popular Machine Learning tool from the list below and explore the features within that tool, using the dataset used above
   a. PyTorch
   b. Scikit-learn
   c. Keras
   d. TensorFlow
3. In a whole class discussion, discuss and explore how to complete the following data processing techniques:
   a. Cleaning

b. Normalising
c. Preprocessing

EXTENDED READING

- Click on *this* to explore more about Machine Learning:

  https://developers.google.com/machine-learning/crash-course/
- Click on *this* to learn more about Python conditional statement:

  https://openstax.org/books/introduction-python-programming/pages/4-2-if-else-statements

Review Questions

1. What function in Python is used to convert a string input into a decimal number?
2. What does it mean for training data to be biased?
3. What is fairness in machine learning?
4. Explain how the base case in a recursive function prevents infinite loops.
5. Why is it important to use data from diverse groups when training a machine learning model?
6. Why is it important to anonymise data when building a machine learning model?
7. How can developers ensure accountability when a machine learning model makes unfair decisions?
8. What is the difference between supervised and unsupervised learning in terms of data usage?
9. How would you use unsupervised learning to group customers based on their shopping behaviour in a supermarket?
10. Write a program to check if a player's age is within a certain range to qualify for a competition.
11. Write a program to determine if a price is under a budget.
12. Write a program that uses a for loop to count down from 10 to 1.

SECTION

6

USE OF WEB PAGE EDITORS

COMPUTATIONAL THINKING (PROGRAMMING LOGIC)

Web Technologies and Databases

INTRODUCTION

This section will help you learn how to use tools for creating web pages. You will explore how to design and organise web pages with sections, text, images, and multimedia. You will also learn about important topics like web page editors, Hyper Text Markup Language (HTML), Cascading Style Sheets (CSS), page layout, and formatting. These skills will help you build strong programming knowledge and solve challenging computer problems.

Key Ideas

- **Database:** A place to store and organise information.
- **Key:** A unique identifier in a database that ensures no two records are the same.
- **Queries:** Statements used to get information from a database.
- **SQL:** A special language used to query a database.
- **Relations:** Connections between tables in a database.
- **Entity:** An object or thing in a database.
- **Attributes:** A characteristic of an entity.
- **Editors:** Software used to write code, create and manage websites.
- **HTML:** The basic code used to structure and design web pages.
- **CSS:** A language used to style web pages.

WEB DEVELOPMENT

To start with web development, familiarise yourself with this simple website in **Activity 6.1**.

Activity 6.1

(Review) Visualising a website

1. Individually, observe and make notes on Ghana government's website by visiting *https://www.ghana.gov.gh/* or clicking *here*.
2. Share with the class what you have observed.

Web development is the process of making websites and web applications that people can use on the internet. It includes writing code, designing how the website looks and arranging the content to create pages that work well and are easy to use. Websites can be simple with just plain text or very advanced (like *Ghana Government Website*, *Facebook*, *X*, *Instagram* or online shopping sites like *jumia.com.gh*, *tonaton.com*, etc).

Importance of Web Development

Web development has improved our daily lives due to the following reasons:

1. **Accessibility**: Websites provide a platform for sharing information and services with people across the globe. The Ghana Education Service (GES) *website* allows students, teachers, parents and the public to access official updates and resources.
2. **Business Presence**: Having a website for your business allows the business to connect with a larger audience and showcase your products or services. *Jumia Ghana* and *Melcom online* are examples of such businesses, reaching customers beyond its physical locations.
3. **Communication**: Blogs, forums and social media on websites and web apps make it easier for people to interact and share ideas. *GhanaWeb* provides news, forums, and blogs for Ghanaians to stay informed and engage in discussions.
4. **Education**: Schools and educational organisations use websites to share resources and information with students and teachers. *WAEC Ghana* provides a website that helps BECE and WASSSCE graduates like you to check your exam results and access educational materials.
5. **E-commerce:** Online shopping platforms help businesses sell their products directly to customers via the internet. *Jumia Ghana* enables customers to shop for a wide variety of products and have them delivered to their doorsteps.

Types of Websites (Static vs. Dynamic)

It is worth knowing that we have two types of websites. These are static websites and dynamic websites. Let us look at them in details.

Static Websites

Static websites have fixed content, meaning the information on each page stays the same for everyone who uses it.

Characteristics of Static Websites

There are some characteristics a website needs to possess to be classified as a static website. These features include the following:

1. Content does not change unless manually updated by the developer: The information on the website stays the same until someone, like a web developer, changes it.
2. Easier and faster to create and host. Static websites are simpler to build and can be published on the internet more quickly.
3. Generally cheaper to develop and maintain. Static websites do not require complex features hence are often less expensive to create and keep running.
4. Suitable for small websites with limited content updates, such as portfolios or informational websites. Static websites are ideal for smaller sites that do not need frequent changes like a personal portfolio or a business information page.

An example is a small shop's website that lists products and contact details of the business without needing frequent updates.

Dynamic Websites

Unlike static websites which have fixed contents, dynamic websites are more advanced and interactive. It interacts with users and suggests or recommends products due to past experiences with the sites. Server-side technologies, such as PHP, ASP.NET, or Node.js, along with databases, are used to create content that changes depending on user interactions or other factors. An example of such a website is the JiJi Ghana website we looked at in our introduction lesson on web development, where the contents, such as product listings, prices, and recommendations change based on what a customer searches for or clicks on. This type of website is used when regular updates and personal interaction are needed to keep the content fresh and relevant for users.

Characteristics of Dynamic Websites

Also, there are some characteristics a website needs to possess to be classified as a dynamic website. These features include the following:

1. Content can change based on user inputs, time, or other variables: Dynamic websites adjust the information they show, based on what users do on the site, the time of day, or other conditions.
2. More interactive and engaging for users: Dynamic websites allow users to interact more. For instance, when you scroll through posts on a social media site or add an item to your shopping cart, the website responds to what you are doing, making it feel like you are in control.

3. Can handle large amounts of data and frequent content updates: Dynamic websites can handle a lot of information and update it easily. For instance, an e-commerce website might have thousands of products, and when an item goes out of stock or a new product is added, the website automatically updates to reflect these changes without needing someone to do it manually.

You can now distinguish between static website and dynamic website based on your understanding of the content so far.

Activity 6.2

Comparison between a Static Website and a Dynamic Website

1. Individually, based on your understanding of static and dynamic websites, compare them using the following features:
   a. content update
   b. interactivity
   c. complexity
   d. cost
   e. development time
2. Present your comparison in a table.
3. Share your response with peers for discussion in class.

Activity 6.3

Identifying websites by type

1. Individually, observe Ghana government's website by clicking *https://www.ghana.gov.gh/* or *here*.
2. In pairs, compare it with *Jumia Ghana's website* using the link *jumia.com.gh*.
3. Identify at least three key differences in functionality and interactivity.
4. Discuss whether each site is static or dynamic, with your reasons for classification.
5. Share your discussion with the class.

Role of Databases in Web Development

Databases play vital roles in web development. Typically, databases are used to store and manage data for web applications. They enable web developers to save, access and modify information whenever required. For example, on an e-commerce platform, its

database stores details such as product names, prices, and customer orders. When you browse or search for items, the website fetches this information from the database and displays it to you. If you make a purchase, the database updates to reflect the new order and adjust stock levels accordingly.

Types of Databases

Databases are grouped into two types, namely Relational Databases and NoSQL (non-relational) Databases.

1. **Relational Databases**

   Relational databases store data in tables like a spreadsheet. Each table has rows and columns. Each row is a piece of data (e.g., a customer or a product), and each column represents a property of that data (e.g., the name, price, or address). These tables can be linked together, based on shared information using keys such as a primary key or a foreign key.

   Examples of relational databases include: **MySQL** or **Microsoft SQL Server** are relational databases. For instance, an online store's database will have one table for products and another for customers. The tables are linked, so that the store can check which products each customer has purchased.

2. **NoSQL Databases**

   NoSQL databases are used for storing large amounts of data that cannot fit well into tables. These databases are more flexible and can store data in many formats, such as documents or graphs. They are good for handling big data or data that changes often.

   An example of a NoSQL database is **MongoDB**. It is often used for websites or apps that have lots of data that does not fit neatly into tables, like social media platforms or online games were data changes frequently.

   NoSQL databases are often referred to a non-relational, as the data held inside is unstructured and does not have any direct relationship with the other data held in the database.

Introduction to Web Page Editors

Web page editors are software tools that help people create, edit, and manage the content and design of websites. They are easy to use, making the process of building websites simpler for beginners, while also helping experienced developers work faster.

Purposes of Web Page Editors

There are reasons one will use a web page editor. Some of these reasons are:

1. **Makes web development easier:** Web page editors have tools that make it simple to create and edit web pages, even if you are new to web design.

2. **Streamline Workflow:** They bring together tasks like writing code, designing the layout, and fixing errors into one platform, making work faster and more organised.
3. **Enhance Productivity:** Features like colour-coded text (syntax highlighting), suggestions for writing code (code completion), and drag-and-drop tools help you work faster and complete your tasks quicker.
4. **Ensure Compatibility:** These editors allow you to easily test websites, to ensure they work properly on different browsers and devices, like mobile phones and computers.

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**.

1. **WYSIWYG Editors:** Allows developers to design web pages by simply dragging and dropping elements like text, images, and buttons onto a screen, without needing to write any code. The editor automatically generates the HTML, CSS, and JavaScript code in the background (this is typically referred to as low or no code development).

   Features

   a. **Drag-and-Drop Interface:** Users can easily add and move items like text, images, or buttons on the web page by dragging and dropping them.

   b. **Real-Time Preview:** As you make changes, you can instantly see how the final web page will look, making it easier to adjust designs.

   c. **Templates and Themes:** These editors come with pre-designed templates and themes that you can customise to suit your needs, thereby saving time and effort.

   d. **Integrated Tools:** These editors often include extra tools for managing pictures, videos, and other multimedia content used on the website.

   Examples of this WYSIWYG

   a. **Adobe Dreamweaver is a** powerful editor with tools for beginners and advanced users.

   b. **WordPress is a** content management system (CMS) platform with a WYSIWYG editor for creating and managing website content.

   c. **Wix and Squarespace** are online platforms that let you design and host websites using simple drag-and-drop features.

2. **Code Editors** are tools designed for developers who want to write their own code manually. They provide a text-based platform where developers can create websites using programming languages like HTML, CSS, and JavaScript. These editors include advanced features that make it easier to write, edit, and manage code effectively. A developer might use software such as **Visual Studio Code** or

Sublime Text to manually write the code for a website, giving them full control over its structure and design.

Features of code editors

a. **Syntax Highlighting:** Code editors use different colours to highlight various parts of the code. This makes it easier to read and identify mistakes.

b. **Code Completion:** While typing, the editor suggests or completes the code for you, saving time and helping you to avoid errors.

c. **Debugging Tools:** These tools help find and fix mistakes in your code, ensuring the website works correctly.

d. **Version Control Integration:** Code editors can connect to systems like **Git**, which track changes made to your code. This allows you to manage your changes providing an audit trail and allowing you to go back to earlier versions if something goes wrong.

Examples of this Code Editors

a. **Visual Studio Code**: A popular tool by Microsoft that supports many programming languages and has useful extensions.

b. **Sublime Text**: A fast, lightweight editor known for being simple yet powerful.

c. **Atom**: An open-source editor made by GitHub, loved for its flexibility and add-ons.

Activity 6.4

WYSIWYG and Code Editors

1. In your groups of no more than five, write at least three differences between WYSIWYG and Code Editors.
2. Present your findings to your peers

Activity 6.5

Exploring WordPress

1. Visit youtube.com and search for “how to use WordPress” to build a portfolio website for free
2. Follow the instruction in the video and create a portfolio website on your own
3. Share your work with your class for discussion and feedback.

BASIC HTML AND CSS

What is HTML?

HTML stands for HyperText Markup Language. HTML is a language, which makes it possible to present information on the Internet. That is, web pages all over the world consist of HTML. However, HTML is not a programming language but a markup language. The idea behind HTML was born at the European Laboratory for High Energy Physics (CERN) in Geneva, Switzerland, as early as 1989. A year later, the World Wide Web project was also started there. The purpose was to make it easier for scientists at different universities and research laboratories to gain access to research documents of their colleagues.

The logo of HTML is shown in **Figure 6.1**.

Figure 6.1: Logo of HTML

An HTML document is simply a text file (it may also contain images, video or audio) that contains the information you want to publish and the appropriate markup instructions indicating how the browser should structure or present the document.

HTML consists of **elements** and **tags**. We will practice how to use these elements and tags in the subsequent activities.

Markup elements are made up of a start tag, such as ** **, and typically, though not always, an end tag, which is indicated by a forward slash within the tag, such as ** **. The tag pair should fully enclose any content to be affected by the element, including text and other HTML markups.

An HTML **element** consists of both the content and its markup. Elements are identified by tags in the text source. A **tag** consists of the element name (usually an abbreviation of a longer descriptive name) within angle brackets (**< >**). The element name appears in the opening or start tag and again in the closing or end tag, preceded by a forward slash (/). The tags added around the contents are referred to as the **markup**. It is important to note that an element consists of both the content and its markup (the start and end tags). Elements that have content are often referred to as **Containers**.

E.g ** ** text ** **. Not all elements have content. These elements are referred to as **Empty elements**. E.g **<input type="text" name="textbox" />**.

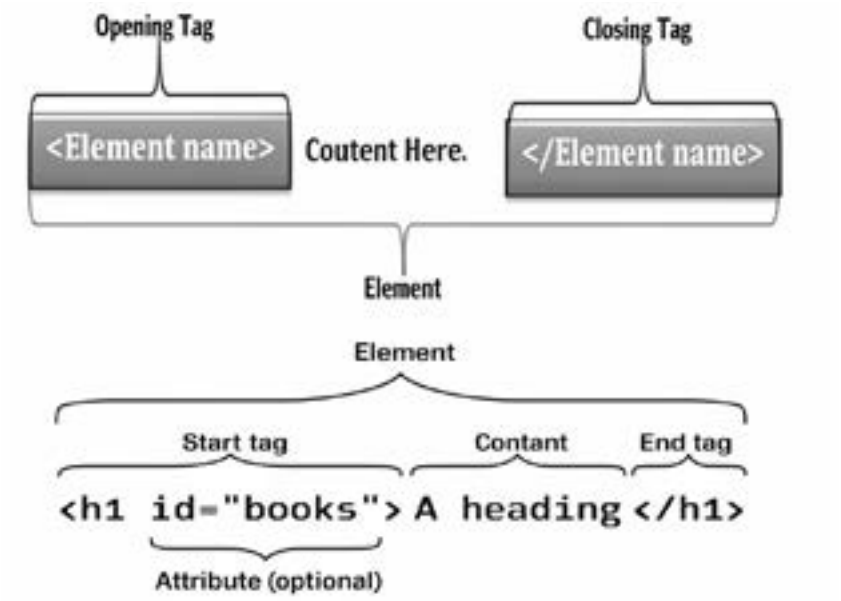


Figure 6.2: HTML Element STRUCTURE

The Basic Structure of an HTML Document

The way a document is marked up with elements and their attributes is according to a Document Type Definition (DTD). DTDs are a set of rules that govern the way in which a document can be marked up. The authoritative source for information about HTML and the HTML DTD is the World Wide Web Consortium(W3C) at *http://www.w3c.org*. An HTML document follows a specific structure with several important elements that help the browser understand how to display the content.

The basic HTML5 document structure is as follows:

<!DOCTYPE html>

<head>

<title> *Title of page goes here***</title>**

</head>

<body>

Main content of the web page goes here.

This is what is displayed in the browser window

</body>

</html>

Let us discuss the structure in more detail.

1. **Document Type Declaration:**

The Document Type, or doctype for short is indicated by the **<!DOCTYPE>** tag in HTML documents. This specifies the rules for the document language, so the browser knows how to interpret the HTML code and display it properly.

2. **HTML Element:**

   The **<html> </html>** tags surround the entire HTML document, and it is referred to as the **root element**. The root element is the container element for all other elements in the document. The opening tag; **<html>** tells the browser that, whatever content that comes after it should be treated as an HTML content until the browser meets the closing tag **</html>.**

3. **Head Element:**

   The **<head> </head>** tags specify the header segment of any HTML document. The **<title> </title>** tags are used within the **<head> </head>** tags to specify the title of a given web page.

   Other tags such as **<meta>, <link>, <style> and <script>** are also used within the **<head>** tag of an HTML document.

4. **Body Element:**

   The **<body>…</body>** tag of a web page play an important role with regards to the page's actual content. The **body** element contains information about the page's background colour, the background image, as well as the text and link colours, and all other items that are displayed on a web page. If the **body** element is left blank, web browsers will revert to their default settings.

   The detailed HTM5 document structure looks like this:

```
<!DOCTYPE html>
<head>
<title> The title of the page goes here</title>
</head>
<body>
```

   Other HTML5 elements and tags are used here within the body tag to produce the actual content of the web page.

```
</body>
</html>
```

Activity 6.6

Creating a Web Page using HTML

1. Create a folder on the desktop named "Programming Class"
2. Create another folder in the Programming Class named "HTML_Lessons"

3. Create a text file using notepad in HTML_Lessons
4. You can name the file as "Lesson1.html"
5. Ensure you have "html" at the end of the file naming, or else it will not be recognised as an html file.
6. Launch the "Lesson1" file with any chosen text editor of your choice (e.g. notepad, VS Code or Sublime Text 3).
7. In the editing area, type the required elements of a basic HTML document as in the example below.

   <!DOCTYPE html>
   <head>
   <title> My First Web Page</title>
   </head>
   <body>
   </body>
   </html>
8. Type any text of your choice within the **<body> </body>** tags of the document structure.
9. Save the file (Ctrl+S).
10. Double click on the file in "HTML_Lesson" folder to preview your page in the browser.
11. Try to identify where the title "My First Web Page" and the content you typed in the body segment appears.
12. Share your web page with the class for discussion and feedback.

All you need to learn HTML is a simple text editor and a web browser and you are good to go. The most important thing here is to learn and understand the various tags that are used within the body segment of the entire document structure.

Common HTML Tags

The commonly used tags for formatting within HTML documents are:

1. **Headings: <h1>** Heading 1**</h1>**

   Headings are used to define titles and subtitles on a page (just like the headings in Word documents). There are six levels of headings which go from from **<h1>** through to **<h6>**, with **<h6>** being the smallest.

2. **Paragraphs: **Paragraph** **

   Paragraphs are used to define blocks of text. They are enclosed in ** **Paragraph** **, tags.

Activity 6.7

Headings and Paragraphs in HTML

1. Create a new file named "Lesson2"
2. Copy the elements below and replace the name within the tags with your name and observe the difference

   <h1> Raphael Senyo Dordoe <h1/>

   <h2> Raphael Senyo Dordoe <h2/>

   <h3> Raphael Senyo Dordoe <h3/>

   <h4> Raphael Senyo Dordoe <h4/>

   <h5> Raphael Senyo Dordoe <h5/>

   <h6> Raphael Senyo Dordoe <h6/>
3. Use the paragraph tag to write a simple message about yourself like

   <p>Hi, I'm Sir Raphael, and I'm passionate about creating safe and inclusive learning spaces for my students by using a variety of classroom management techniques.

   One way I do this is by providing more opportunities for learners from underprivileged groups.

   For example, my school doesn't have computers for practical work, so I improvised with my personal laptop. I encouraged a girl who had never used a desktop or laptop before by making her feel safe and confident to practise her code skills in HTML before the whole class. **<p/>**
4. Share your work with the class for discussion and feedback.

Before we continue writing more code, watch this short video (*https://youtu.be/8xMnkt474q4*) about indentation in programming, to refresh your mind.

Activity 6.8

HTML Structure

1. Duplicate the "Lesson1" HTML file and name it "Lesson3"
2. Copy the code from "Lesson2" and paste it into the body segment of "Lesson3".
3. Save the file and launch "Lesson3" in a web browser
4. Write your observations between "Lesson2" and "Lesson3", what are do you notice that is different?
5. Show your work to your teacher

It is important to note that the structure of "Lesson3" is the best way to write your HTML code to keep your work organised.

Activity 6.9

HTML tags

1. Search for "HTML tags" on the internet
2. Explore how they are used
3. Take key notes of the following, we will use them later to create a personal portfolio.
   a.
   b. <hr>
   c. <img>
   d. <video>
   e. ...
   f. <header>...<header/>
   g. <footer>...<footer/>
4. You can practise your findings by creating in a new file named "Lesson4" and using some of the HTML tags that you have researched

Links

HTML links are hyperlinks, meaning that when you click on a link, it takes you to another document, page or web resource and when you move the mouse over a link, the mouse arrow will turn into a little hand. Links are defined with the **<a>** tag and require an **href** attribute to specify the target URL.

The syntax for inserting links is *link text* .

For example: Visit T-TEL.

In this link, the target URL is *https://t-tel.org/* and the text visible to users of the site to click on is "Visit T-TEL". Let's put this into practise for a better understanding.

Activity 6.10

HTML links

1. Duplicate "Lesson2" and rename it as "Lesson4"
2. Add a short paragraph about education in Ghana using the <p> tags.
3. Write links to the websites of the following institutions using <a> tags.
   a. *https://moe.gov.gh/* (The user should see "visit MoE").
   b. *https://ges.gov.gh/* (The user should see "visit GES").
   c. *https://t-tel.org/* (The user should see "visit T-TEL").

4. Open one of the links in a new browser window.
5. Share your result with your peers.

Images

Images or pictures can be inserted into HTML documents. It is done using the ***<img >*** tag.

a. The ***src*** (Source) attribute is the only required attribute of the ***<img >*** tag. This attribute tells the browser where to find the image that can be inserted into the web page.

b. The ***alt*** attribute provides a text-based description of the image (alt is important because it helps visually impaired users understand what the image is about, making the webpage more accessible).

c. Other attributes of the ***<img >*** tag include ***height, width*** *and* ***align*** *which will be explored later.*

The <img> tag in HTML is used to embed images into the web pages content. It is a self-closing tag, so there is not need to provide a closing tag at the end. Below is an example of the <img> tag in use:

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

The above code contains the following attributes:

- src: this specifies the location of the image. It can be a relative path (e.g. 'image/photi.jpg) or it can be an absolute URL (e.g. 'example.com/photo.jpg)
- alt: this provides alternative text for the image if it cannot be display on the web page. This is important for both accessibility and SEO (search engine optimisation)
- width: this defines the width dimension for the image, it can be defined as either pixels or a percentage value
- height: this defines the height dimension for the image, it can be defined as either pixels or a percentage value

Activity 6.11

HTML images

1. Duplicate "Lesson4" and rename it "Lesson5"
2. Download four pictures of your schools into a new folder named "images" in the "HTML_Lessons" folder
3. Write the header as <h1> Pictures of my school </h1>
4. Write the image code as <img src=" " alt=" ">
5. Right click on the first image and click on copy as path

6. Paste the path into the quotes of src.
7. Write a short description of the image in the alt attribute.
8. Name the image using the ... tags
9. Repeat steps 4-9 for the other three pictures
10. Share your result with your peers.

Lists

HTML supports **ordered**, **unordered** and **definition** lists. It is also possible to nest one type of list within another in HTML documents.

1. Ordered List

An ordered list is a list in which the list items are numbered sequentially. Ordered lists are enclosed in the ** ...
** tags. Each list item is placed within the ** ...
** tags using the ** ...
** tags.

For example

<h3>My favourite dancehall icons in Ghana:</h3>

 Stonebwoy

 Shatta Wale

 Samini

 Rocky Dawuni

 Shasha Marley

2. Unordered List

An unordered list is used when the list of items does not have any necessary sequence of appearance. It is also called a **bulleted list**. Unordered lists are enclosed in the **
... ** tags. Each list item is placed within the **
...
** tags using the **
... ** tags.

For example

<h1> The following are my favourite foods:</h1>

<ul>

 Fufu

 Akple

 Kenkey

Red Red

TZ

**
**

3. Definition Lists

This is used to present a term and its definition formatted like a glossary or a dictionary. It is the ideal format to present lists of words or phrases with their meanings. The tags used for a definition list are <dl> … </dl> with both the term defined as the <dt> tag and the description as defined as the <dd> tag, note that both the term and the definition are needed for each list item in the <dl> … </dl>

<dl>

<dt> Definition Term </dt>

<dd> Definition Description </dd>

</dl>

Activity 6.12

Creating a List of Favourite Meals

In groups of no more than five,

1. Create a new file named "Lesson6"
2. Create an ordered list of five local foods that you enjoy the most.
3. Create an unordered list of six local football clubs that you know in Ghana.
4. Create a definition list of three words
5. Share your list with your peers.
6.

Activity 6.13

Glossaries

In groups of no more than five,

1. Create a glossary of five terms you have learned in HTML in a new file named "Lesson7".

2. Share your glossary with your peers.

Table 6.1: Summary of List Elements

Tag	Description
	Defines an unordered list
	Defines an ordered list
	Defines a list item
<dl> </dl>	Defines a description list
<dt> </dt>	Defines the term in a description list
<dd> </dd>	Defines the description in a definition list

Tables

Tables provide an excellent way to organise and display information on web pages. Tables are defined using the **<table>…</table>** tags. The HTML **<table> element** contains the table information, which consists of **table header elements (<th>)**, **table row elements (<tr>)**, and individual **table data cells (<td>)**. The letters **td** stand for "table data" which is the content of a data cell. A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, and so on.

These are the three elements that are used most frequently when you are building tables.

Table 6.2: HTML Table Elements

Element	Description
<table>…</table>	Establishes the table. It contains all other elements that specify captions, rows and the content.
<tr>…</tr>	Table row. It contains rows of table cells.
<td>…</td>	Table data cell. This contains the actual table data held in each cell.
<th>……</th>	Table header cell: It contains the header information for a column of data.
<caption>…</caption>	Provides a short description of the table's contents.
<thead>…<thead>	Signifies table header.
<tbody>…<tbody>	Signifies table body

Element	Description
<tfoot>…<tfoot>	Signifies table footer
<col />	Specifies column properties
<colgroup>…</colgroup>	Specifies multiple column properties

Activity 6.14

Tables in HTML

1. Copy and run the following code in a new file named "Lesson8".

```
<table>
<tr>
<th>Title of Music </th>
<th> Artiste </th>
</tr>
<tr>
<td> Forward to Africa </td>
<td> Joseph Hill </td>
</tr>
<tr>
<td> China Roses </td>
<td> Enya </td>
</tr>
<tr>
<td> Floral Street </td>
<td> Enya </td>
</tr>
<tr>
<td> Humble Africa </td>
<td> Joseph Hill </td>
</tr>
<tr>
<td> Nah Stay Inna Babylon</td>
<td> Joseph Hill </td>
</tr>
<tr>
<td> Miss Independent </td>
```

```
<td> Neyo </td>
</tr>
<tr>
<td> Storm in Africa</td>
<td> Enya </td>
</tr>
<tr>
<td> Count on Me</td>
<td> Whitney Houston</td>
</tr>
</table>
```

2. Edit the opening tag **<table>** to **<table width= "50%", border = "2">**
3. Observe what changes this makes and write down your observations

Activity 6.15

Creating a Personal Timetable

In groups of no more than five,

1. Create a personal study timetable with the days of the week as the header row in a new file named "Lesson9".
2. Add three rows for subjects.
3. Specify the subject and the time for each of the three subject rows.
4. Share your table with your peers.

Introduction to CSS

What is CSS?

CSS is a styling language that defines how the website should look like. CSS is an acronym which means Cascading Style Sheets. *CSS* is the recommended way to control the presentation layer in a web document. The best way this is done is by creating effective and efficient style rules. A style sheet is a grouping of formatting instructions that can control the appearance of many HTML pages at once.

CSS is a language that defines the style constructs such as fonts, colours and positions, which are used to describe how information on a web page is formatted and displayed.

Benefits of CSS

CSS offers several significant benefits. These benefits include:

1. Apply the same styles across multiple pages.
2. Separate styling rules, which can be used for different output media.
3. Easily change the appearance of your website without modifying HTML content.
4. Keep the structure (HTML) and styling (CSS) separate for better organisation.
5. All styling is kept in a limited number of style sheets.
6. There is some saving on bandwidth, therefore making the webpage(s) load faster.
7. The separation of content from presentation makes it easier for site owners to reuse the content for other purposes, such as RSS feeds or text-to-speech conversions.

CSS Syntax and Selectors

A rule or "rule set" is a statement that tells browsers how to render elements on an HTML page and it consists of a **selector** followed by a **declaration** block. Each declaration consists of a property and a value. A CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly brackets.

The syntax rule looks like the image below.

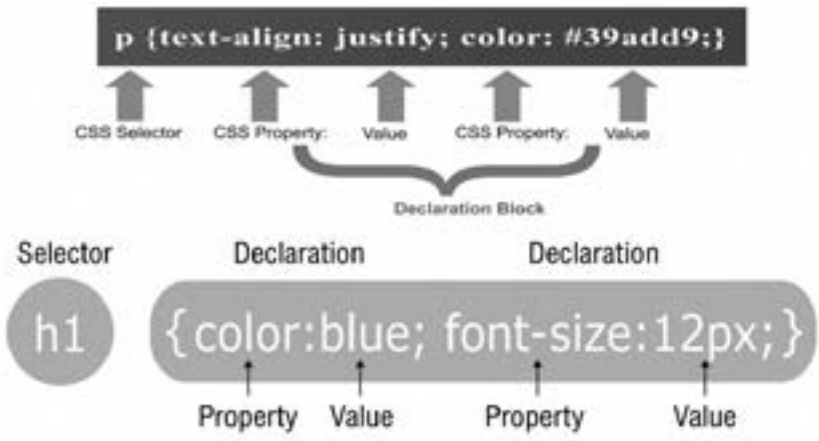


Figure 6.3: The syntax rule

The selector is used to 'select' or 'target' the HTML element that you want the style to apply to. This is how we tell the web browser to apply a particular style to a particular html element. For instance, when you want to style all the paragraphs on your web pages.

- The letter **"p"** would be our selector and it targets (selects) all paragraphs.
- The **declaration block** is everything between the curly brackets. What you see between these curly brackets is referred to as the ***Property*** *and the* ***Value***. This is how we define the properties we want to apply to our web pages.

Combining HTML and CSS

To work in CSS, one must be familiar with some markup languages like HTML or XHTML, so that you can add personalised styles on the markup documents. This is because CSS rules are applied to HTML or XHTML elements. There are three ways to apply CSS to HTML elements.

1. **Using inline CSS**

   An inline style can be used if a unique style is to be applied to one single occurrence of an element. To use inline styles, use the style attribute in the start tag of the relevant element. The style attribute can contain any CSS property. The example below shows how to change the text colour of a paragraph:

   This is a paragraph.

   NOTE CSS use American English spellings, for example 'colour' is spelt 'color'

2. **Using internal CSS**

   An internal style sheet can be used if one single document has a unique style. Internal styles are defined in the <head> section of an HTML page, by using the **<style>** tag, like this:

   <head>

   <style type="text/css">

   body {background-color:yellow}

   p {color:blue}

   </style>

   </head>

3. **Using external CSS**

   An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire web site by using one CSS file. Each page must link to the style sheet using the **<link>** tag. The **<link>** tag goes inside the **<head>** section as shown in the example below:

   <head>

   <link rel="stylesheet" type="text/css" href="mystyle.css" />

   </head>

Activity 6.16

Creating My Portfolio

Using basic HTML and CSS, complete the following

1. Create a new folder named "**CSS Lessons**".
2. Create a new HTML file named "Portfolio.html" in the folder

3. Create a static web page titled "About Me" in your HTML file
4. On your web page, include the following
   a. Your name,
   b. A description about you,
   c. A picture of you,
   d. Your favourite subject, food, teacher, hobbies,
   e. The name of your best friends,
   f. The name and description of your school,
   g. A picture of your school,
   h. A google map link to your school.
5. Show your work to your teacher to ensure it has the required details in step 4.

Activity 6.17

Styling My Portfolio

1. Follow the instruction in this video
   https://drive.google.com/file/d/1mcVQzsUA3Vr2aedmT5zNSTK4yWd2-H-g/view?usp=drive_link
2. Style your portfolio website.
3. Share your websites with your peers.

Note: The source code is attached in this link, you can download, edit and improve it.

a. Html
   https://drive.google.com/file/d/15vRumOyrnizJsfd1cAXlLBDIZV8qMC/view?usp=drive_link
b. CSS:
   https://drive.google.com/file/d/1jDsdnKHoQPkUJggz03JfUxPFERwbCRSv/view?usp=drive_link

IMAGE AND VIDEO INTEGRATION

Videos

The <video> tag lets you include video files directly from your server. It supports various video formats like MP4, WebM, and Ogg.

<video controls width=”width” height=”height”>

<source src=”URL” type=”video/format”>

Your browser does not support the video tag.

</video>

The above code contains the following attributes:

a. **controls:** this adds playback controls (play, pause, volume) to the video player
b. **width:** this defines the width dimension for the video player, it can be defined as either pixels or a percentage value
c. **height:** this defines the height dimension for the video player, it can be defined as either
d. **source:** this specifies the video file’s path and format

Integration of Video and Audio files to a website

1. Video files are a collection of images usually with a related sound file and can be inserted into the HTML. Some common video formats are: MPEG (.mpeg/.mpg), AVI for Audio/Video Interleave (.avi), and QuickTime (.qt/.mov), etc.
2. Integrating audio into your web pages can enhance the user experience by providing sound effects, background music, or narrated content.

Activity 6.18

Adding Video and Audio Files

1. Search online for how to add video and audio files respectively to an HTML page
2. Follow to instruction or demonstration to add a video file to your portfolio page
3. Share your results with the class for discussion and feedback.

Note: there are other interactive multimedia files like, slideshows and maps, which can be integrated in a website using specific plugins.

MULTIMEDIA INTERGRATION

Best Practices for Multimedia

There are things that need to be done and considered when developing a website. It is prudent to follow such standards to write syntactically correct code and get websites that offer the best in terms of the content loading efficiently, looking good, and being accessible to all users. These best practices include:

Optimising Media for the Web

Optimising media means making sure that the images and videos on your website load quickly and look good, no matter what device or internet speed people may be using. This is important because slow websites can frustrate users and make them leave. Good optimisation improves the user experience and makes websites accessible to everyone. The following need to be optimised for a website.

1. Image Optimisation

Images often take up the most space on a website, so optimising them is essential to make your website load faster. This involves choosing the right format and reducing the file size, while keeping the quality high. Examples of file types for images include:

a. **Joint Photographic Expert Group (.JPEG)**: Best for photos and images with lots of colours and detail. Compresses the image to make the file smaller, which helps the website load faster.

b. **Portable Network Graphic (.PNG):** Ideal for images with transparent backgrounds or high-quality graphics, like logos or illustrations. Does not lose quality when compressed but creates larger file sizes compared to JPEG.

Compressing Images

Image compression reduces file size without making the image look blurry. Smaller files load faster on websites. Tools to compress images include:

i. *TinyPNG*: Reduces the size of PNG and JPEG images.

ii. *JPEG-Optimizer*: Helps you compress JPEG images while keeping them clear.

2. Video Optimisation

Videos are another type of media that can slow down a website. Optimising videos means choosing the right format and reducing the file size while, keeping good quality.

a. **Choosing the Right Format:**
   i. MP4: This is the most popular video format because it balances file size and quality. MP4 works on almost all devices and browsers, so it is a safe choice for websites.
   ii. WebM and Ogg: These formats offer better compression than MP4, making the file sizes smaller. However, they are not supported by all browsers, so you might need to provide an alternative format.

3. Compressing Videos:

For video compression, use tools like *HandBrake* **and** *Clipchamp* to reduce video sizes, while keeping high quality for web use.

INTRODUCTION TO RESPONSIVE DESIGN

Responsive design means creating websites that can change their look and layout to fit different devices such as computers, tablets, or smartphones. This makes sure that anyone can use the website easily, no matter the size of their screen. Websites that adapt to the device so that it fits the size of the screen are called responsive websites.

Importance of Responsive Design

Below are some of the benefits that users get from responsive website design:

1. **Improved User Experience:** A responsive website makes it easier for users to find information or complete tasks, whether on a big screen or a small one.
2. **Increased Mobile Traffic:** Many people use mobile phones to browse the internet. Responsive websites make sure mobile users can access everything easily, without distortion or excessive scrolling.
3. **Cost-Effective:** Instead of creating separate websites for phones and computers, a responsive design works across all devices
4. **Search Engine Optimisation (SEO) Benefits:** Search engines like Google prefer responsive websites, so they appear higher in search results. Google recommends responsive web design as the best practice for mobile configuration.
5. **Futureproofing:** A responsive website will still work as new devices with different screen sizes are developed. This makes sure that the website will fit any future device that may be developed. This makes responsive design beneficial to users.

Key Principles of Responsive Design

1. **Fluid Grid Layouts:** A website's layout should adjust based on the screen size. For example, instead of fixing a section to 300 pixels wide, make it take up 50% of the screen. If a webpage is divided into two halves, each half will resize depending on the screen size.

Figure 6.4: Responsive Grid

From **Figure 6.4**. You will notice that the display layout on a desktop, laptop screen is different to that of the mobile screen. This is achieved by setting the display width in percentages and not keeping the widths fixed with the use of divs.

Activity 6.19

Exploring Responsive Grid

1. Copy and paste the code in a code editor and RUN it.
2. Reduce the page width by dragging the left or right side to make it smaller
3. Observe how the items rearrange because of the layout created.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Fluid Layout Grid Example</title>
<style>
    .container {
      display: grid;
       grid-template-columns: repeat(auto-fit, minmax(100px, 1fr));
        gap: 10px;
        padding: 10px;
```

```
        }
        .item {
            padding: 20px;
            text-align: center;
            border: 1px solid #ccc;
        }
    </style>
    </head>
    <body>
      <div class=”container”>
        <div class=”item” style=”background-color: yellow”>Item 1</div>
        <div class=”item” style=”background-color: green”>Item 2</div>
        <div class=”item” style=”background-color: orange”>Item 3</div>
        <div class=”item” style=”background-color: blue”>Item 4</div>
        <div class=”item” style=”background-color: pink”>Item 5</div>
        <div class=”item” style=”background-color: gray”>Item 6</div>
        <div class=”item” style=”background-color: purple”>Item 7</div>
        <div class=”item” style=”background-color: red”>Item 8</div>
        <div class=”item” style=”background-color: lightSeaGreen”>Item 9</
    div>
        <div class=”item” style=”background-color: tomato”>Item 10</div>
       </div>
    </body>
    </html>
```

2. **Flexible Images:** Pictures on a website should shrink or expand to fit the screen without breaking the design. Use CSS properties like “max-width: 100%;” to ensure the images resize within their containing elements.

   For example, the CSS code img {max-width: 100%; height: auto;} sets the maximum width of the image to 100% and sets the height to automatic for every device.

Activity 6.20

Exploring Image flexibility

1. Download the index file (*https://drive.google.com/file/d/1kXzcLruOhN8QFH0QbMLvXUrzw3lbV4yQ/view?usp=sharing*)
2. Follow the instructions in the **TODO**

3. **Media Queries:** Media queries are special styles (rules) used to change the look of a website for different screen sizes, like the screen width, height, orientation and resolution. This allows you to create specific styles for different devices. Media Queries will be explored later.
4. **Responsive Typography:** Text should be easy to read on all devices. Use relative sizes like em or rem instead of fixed sizes like px. For instance, the CSS style: body {font-size: 2em; line-height: 1.8;}.

   Note that Responsive Typography will be explored later.
5. **Touch-Friendly Design:** Buttons and links should be large enough for easy tapping on touchscreens. Avoid hover effects, as they do not work well on mobile devices.

Extension Activity 1: Comparing Responsive and Non-Responsive Websites

Instructions

1. **Visit Two Sample Websites**

   Responsive Site: Visit a website known for its responsive design, such as [*https://www.example.com](https://www.example.com*). Resize the browser window to see how the layout adjusts to different screen sizes.

   Non-Responsive Site: Visit a website that is not designed to be responsive. Use [*https://www.nonresponsive.com](https://www.nonresponsive.com*) as an example. Resize the browser window and observe how the site does not adapt to different screen sizes.
2. **Analyse the Differences**

   Layout Changes: How does the layout of the responsive site adjust as you resize the browser window? How does the layout of the non-responsive site change?

   Image Scaling: Observe how images are handled in both sites. Are images scaling properly on the responsive site?

   Text Readability: Compare text readability on both sites as you adjust the screen size. Is text readable on the responsive site across different sizes?

   Navigation: How does navigation change on smaller screens? Is the responsive site's navigation user-friendly on mobile devices?
3. **Document Your Findings**

   Write a summary of your observations. Include specific examples of how the responsive site's design benefits users compared to the non-responsive site.

   Engage in a class discussion to discuss any potential improvements that could be made to the non-responsive site.

WEB ACCESSIBILITY

Web accessibility is about making websites easy to use for everyone, including people with disabilities. For instance, someone with poor eyesight should be able to read the text on a website and someone who cannot use a mouse should be able to navigate using a keyboard. 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:** All users should be able to see or hear the information on a website. This comprises of providing text alternatives for non-text content, making content adaptable and ensuring that the content can be both seen and heard. For example, images should have text descriptions for people who cannot see them, this will often be read out by a screen reader.
2. **Operable:** Users should be able to navigate the website in different ways, like using a keyboard instead of a mouse. For instance, we should be able to use the tab key on the keyboard to move around the different elements of the webpage.
3. **Understandable:** The website should be easy to read and use. Avoid complicated designs or unpredictable behaviour.
4. **Robust:** Websites should work well with different devices and assistive tools like screen readers and future technologies that will emerge. To do this, you need to use valid HTML and follow best practices in developing your website to ensure compatibility and accessibility.

Importance of Web Accessibility

1. **Inclusivity:** This allows everyone, including people with disabilities, to access and use websites. Accessibility ensures that people with disabilities have equal access to information and services, promoting inclusivity and equal opportunity.
2. **Legal Compliance:** Some countries have laws requiring websites to be accessible. 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 and does not violate the fundamental right of the users.
3. **Enhanced User Experience:** Accessibility improvements, like clear navigation, make the website easier to use for everyone and not just those with disabilities.
4. **Broader Audience:** Accessible websites can be used by more people, including those with challenges such as a broken arm. By doing this and making your website accessible, you could reach a larger audience.
5. **SEO Benefits:** Search engines reward accessible websites with higher rankings. They are awarded because they follow best practices for structure and content, making it easier for search engines to index and rank them.

Activity 6.21

Evaluating a Website for Accessibility

1. Copy the GES website URL *https://ges.gov.gh/* and paste it into the address box of your web browser
2. Press **Enter** or click on the forward arrow to check the website's accessibility
3. Observe the result on the left panel
4. Scroll down to see where the accessibility issues where highlighted
5. Write a brief report, descripting the issues you find and suggest ways to make the website more accessible
6. Share your report with your peers

Activity 6.22

Research Assignment on Website

In groups of no more than five, using the internet or online resources and your understanding of concepts of website, complete the following

1. Discuss and document what you think of the following questions.
   a. Why do you think Google prefers responsive designs? How could this impact your business?
   b. Why is it cost-effective to use responsive design and ensure accessibility together for your business? Discuss with examples from local or global websites.
   c. How can you convince a local business owner to invest in a responsive and accessible website? Write a persuasive paragraph.
2. Create a responsive website to show your responses to the above questions.
3. Share the link to your website with your teacher and peers, to get feedback.

CREATING A SHOPPING CART

A **shopping cart** is like a basket you use when shopping online. It helps you pick the items you want to buy, shows you what you have chosen, and lets you pay for them when you are ready. Think of it as your helper when shopping on websites like kromonline.com.

It is an essential component of any e-commerce website. It allows users to select products, view their selected items and proceed to the checkout just like Jumia. See **Figure 6.5** for a typical image of a shopping cart.

Figure 6.5: Shopping Cart

Basic Structure of a Shopping Cart

The structure of a shopping cart typically includes the following:

1. **Cart Container:** This is the main box where all your selected items are placed. Take it as a basket on a website, where you can see all the things you have picked. This is just like the normal basket where you keep your chosen items when you visit Kumasi City Mall, Melcom or Shoprite. The cart is virtual, while baskets are physical to the user.
2. **Item List:** This shows you details about the items you have added. The details may include the name, quantity and price of the item(s).
3. **Cart Controls:** These are buttons that you use to modify your item(s). You can increase or decrease the quantities, remove items and proceed to checkout.
4. **Summary:** A summary section displays the total price and other relevant information. This is the final section where you see the total amount of money you need to pay for everything in your cart. For instance, if your cart has two T-shirts at GH¢580.99 each and a pair of shoes for GH¢358.99, the summary will show: Total: GH¢1,520.97.

Activity 6.23

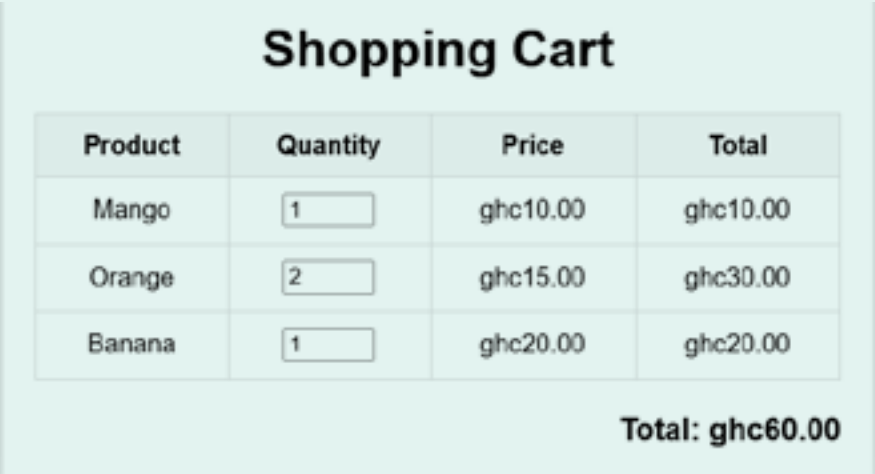
Creating a Shopping Cart Structure

In groups of no more than five, chose a local company (or create a fictional brand) and create an online e-commerce website. You will be competing with the other teams in your class for who had created the best-designed website.

You must use HTML and CSS to design your website, and you must include the following elements on your website:

- Headers
- Paragraphs
- At least one table
- At least one image
- At least one video
- An e-commerce shopping cart to allow customers to purchase items

Below is an example of what a shopping cart looks like, you can choose to model that layout, or you can create your own



Present your website to your teacher and peers for discussion and feedback.

PAYMENT GATEWAY INTEGRATION

A payment gateway is like a digital cashier. When you buy something online, the gateway takes your payment details, sends this information securely to the seller's bank and makes sure the payment is accepted before telling both you and the seller. The simple way to understand gateway is to think of it as a bridge between the buyer (you) and the seller to ensure the payment is safe, successful and fast.

The importance of a Payment Gateway

Payment gateways are very important components of online payment systems. They securely transmit transaction information between a customer's and merchant's banks. Some the importances of payment gateway include:

1. **Security:** Your card details are kept private and safe using special techniques like encryption.
2. **Convenience**: You can use many payment methods like bank cards and mobile money.
3. **Efficiency:** Payments are processed in seconds with quick alert.
4. **Compliance:** They follow strict rules to protect both the buyer and seller from fraud. They follow standards such as PCI-DSS (Payment Card Industry Data Security Standard).

Examples of popular Payment Gateways in Ghana

1. **Hubtel** is a locally built payment platform that supports mobile money (MTN MoMo, AT Money, Telecel Cash), bank cards, and QR codes. Many businesses like online shops use it for seamless integration with their online stores and physical shops.
2. **ExpressPay** is a Ghanaian payment gateway for mobile money, card payments, and bank transfers. It provides an app for utility bill payments, such as school fees and online shopping.
3. **Slydepay** supports payments via mobile money, bank cards, and QR codes. It offers simple integration for small businesses and online stores in Ghana.
4. **iPay** focuses on simplifying mobile money and card payments for businesses. Businesses can both receive payments and manage transactions using its platform.
5. **Paystack** is a popular payment gateway across Africa, including Ghana, which allows businesses to accept card payments, mobile money, and bank transfers. It is known for its smooth onboarding process and developer-friendly tools.
6. **Flutterwave** operates in multiple African countries, including Ghana. It supports mobile money, card payments, and bank transfers. It offers businesses the ability to accept payments from customers across Africa, as well as globally.
7. **MTN Mobile Money API** specifically designed for MTN MoMo users, allowing merchants to directly integrate mobile money payments into their platforms. MTN MoMo is widely used in Ghana making this API highly relevant for local businesses.
8. **Zeepay** enables businesses to accept remittances, mobile money, and card payments. It focuses on linking international remittances to local mobile money wallets.
9. **Global Accelerex** provides secure payment solutions for online businesses, including card and mobile money integrations. It offers tools to streamline payments and reconciliation.

Activity 6.24

Integrating a mock payment gateway

In your, extend your e-commerce website from **Activity 6.23** to include a payment gateway

1. Follow the instructions in the tutorial provided by your teacher (Link to tutorial for integrating a mock payment gateway)
2. Integrate a mock payment gateway, using a sandbox environment available here (Link to either paypal.com or stripe.com)
3. Extend your shopping cart so that you can:
   a. Add items

b. Remove items
c. Change the quantities
d. Calculate the totals
e. Add security measures to your e-commerce website, including HTTPS and SSL certificates, to protect the customers data

USER ACCOUNT MANAGEMENT

Enhancing User Experience

User accounts let you personalise your online shopping experience. For example, when you log into an account on a shopping site like Jumia, Melcom Online or portfoliogh.com, you can see your order history, inbox, vouchers, saved items, recently viewed items, recent searches, and quickly check out, without entering your details every time.

Order Tracking

With an account, you can check where your items are after you buy them. For instance, if you order a phone on the Jumia online shop, from the comfort of your house, the website can tell you whether it's "Being Prepared," "Shipped," or "Delivered. E-commerce provides real-time updates on the status of your orders, thereby improving transparency and your satisfaction.

Customer Support

If something goes wrong, your account lets you contact customer support, track your complaints, and get quicker solutions from customer service representatives who can access your profiles, to provide support and resolve issues more efficiently.

Security

Accounts store private information like your email, address, and bank details. With implementation of secured login mechanisms using authentication, there is less risk of unauthorised access and fraud.

Basic User Account Features

Let us look at some basic account features of online system, especially those that deal with sensitive data and finances.

1. Account Creation

Before you can use the features of an account on an e-commerce site, you first need to create one. In creating an account, you must fill a registration form. The registration typically requests your personal details such as name, email address, and a password. To prove you are a human and not a robot, websites often use CAPTCHA tools, such as identifying pictures of cars, fire hydrants, buses, bicycles, bridges, animals or typing distorted letters. You then need to confirm you are not a robot by providing accurate responses to the CAPTCHA questions.

CAPTCHA

CAPTCHA stands for **Completely Automated Public Turing Test to Tell Computers and Humans Apart**. It is a test that websites use to make sure the person accessing the site is a real human and not a robot or computer program (bot). Bots often try to create fake accounts or spam websites, and CAPTCHA prevents this by asking questions that are easy for humans but hard for bots. By stopping bots, CAPTCHA prevents cyberattacks, like overwhelming the website with too many requests (DDoS attacks) and ensures that only real humans can access sensitive areas of a website, like registration forms or payment sites. **Figure 6.6** below shows an example of a Login page with CAPTCHA.

Figure 6.6: Login page with CAPTCHA

2. Login

After creating your account, you log in with your email and password. Some websites add extra security by sending a code to your phone for you to enter after your password. This type of login has more than one layer of verification. They have two-way or three-way verification mechanism, generally called either two-factor authentication (2FA) or multi-factor authentication (MFA). They are designed to identify the authorised accessor of the account. If you provide the required login credentials, you are given access, otherwise, you will be locked out. The secure login process includes password hashing/salting and encryption.

Form Structure

A registration form is like a paper form; however, it is online. It allows you to enter your details, such as your username, email and password.

Every registration form has a structure which make it clear and easy to use.

1. **Form Elements**: These are the parts of the form that you fill out during registration. They typically include text fields, email fields, password fields, and submit buttons.
2. **Labels and Inputs**: A label is the name written next to or above a field (e.g., "Email Address") to show what you need to enter and the input is where you type the required information.
3. **Submit Button**: Once you have entered all required information, you need to click the Submit button to send your details to the website, so that it can create your account.

Figure 6.7 shows an example structure of a registration form.

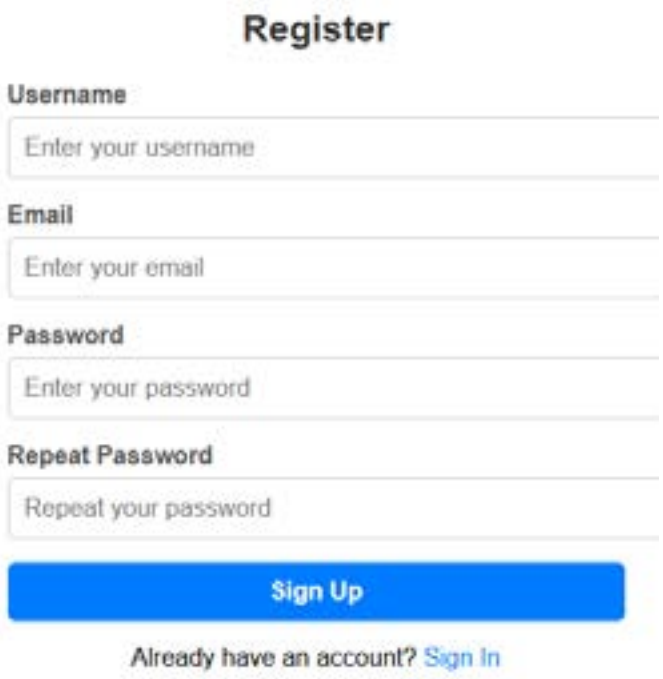


Figure 6.7: Structure of a registration form

Activity 6.25

Design and Fill Out a Registration Form

In groups of no more than five,

1. Search online for "how to create forms using HTML"
2. Pay key attention to the use of
   a. Labels
   b. Names
   c. Requirements
   d. Difference between placeholders and values

3. Create a simple form using the following fields:
   a. Username
   b. Email Address
   c. Password
   d. Confirm Password (to retype and confirm your password)
4. Add a Submit button at the bottom of your form.
5. Do not worry about styling your form at this point but ensure your items are in vertical order.

Managing User Accounts

When you create an account on a website, you also get a **user profile**. This is a special section where you can see and change your personal information.

Profile Management

Profile management enables you to see your username, email, and other details you added when creating your account. If there are changes to your details, like a new email address or password, you can modify your profile and save the changes. This action allows you to manage your profile by updating it in the database.

SECURITY AND TRUSTWORTHINESS

Security is important for websites because they usually collect and store sensitive information, such as name, email, payment information and order history. If this information is not protected, it can lead to serious issues such as identity theft and fraud. To protect one's data on a website, they include integrated layers of security mechanisms that make you trust the site and feel comfortable using it. If a website has poor security, people stop using it, and the company will lose money or even face legal actions.

In Ghana, the Data Protection Act ensures that companies manage personal information safely. If they do not follow these rules, they will face **fines** or other **legal consequences**.

Activity 6.26

Security in E-Commerce

1. Organise yourselves into groups of no more than five. In your groups, discuss and document the following
   a. Why it is important to have a strong password?
   b. What would make you trust an online store?
   c. What can happen if someone else gets access to your personal details?
2. Share your findings with your peers.

When creating an e-commerce website, it is important to follow these three key security principles, to ensure that users' data and the website itself are safe and reliable.

1. **Confidentiality:** Confidentiality ensures that private information, such as usernames, passwords, or payment details is only accessible to the people who are authorised to see it. If you enter your payment details when buying something online, only you and the website should be able to see them — not hackers or unauthorised users. To ensure confidentiality, websites should use secure logins and encrypt sensitive data, so even if it is stolen, it cannot be read.
2. **Integrity:** Integrity means making sure the information on the website is accurate and has not been tampered with by unauthorised people. For instance, if you buy an item worth 100 cedis, someone else should not be able to change it to 500 cedis in the system.
3. **Availability**: Availability ensures that authorised users can access the website and its services whenever they need them. If a customer wants to place an order, the website should be functional and not down due to cyberattacks or technical issues. This also makes the website more reliable.

Secure Authentication Mechanisms

To protect users and sensitive data, e-commerce websites implement secure ways of handling passwords and user logins. They include password hashing and salting. Let us discuss them.

1. **Password Hashing**

   Password hashing is like converting a password into a scrambled code that looks random. This "code" cannot be reversed to reveal the original password. Even if someone steals the hashed passwords, they cannot easily figure out the actual passwords. Imagine you create a password, "Abu1957", the website hashes it into something like "bua75@91dfg#". Each time you log in, the system compares the new hashed version of your password with the stored one to confirm it matches.

2. **Password Salting**

   Salting adds a unique, random string (called a "salt") to your password before hashing it. If two people use the same password, the hashes will still look different because of the unique salt. For instance, for "Abu1957", one user might get "saltA@Uba91#dfg" while the other gets "saltB@xy75#hij".

Implementing Two-Factor Authentication

Two-factor authentication (2FA) adds another layer of security by requiring users to provide two forms of authentication.

In real life, 2FA works this way: the user enters their username and password. After the password is confirmed, the user must provide a second verification code. This code is then sent to their phone, email or generated by an app. If the code is correct, the user gains access. You must note that even if someone steals your password, they cannot log in without the second factor.

Activity 6.27

Securing your password

Organise yourselves into groups of no more than five. In your groups,

1. Discuss the following questions:
   a. Why do you think websites ask for your password twice?
   b. What would happen if you forgot your password after creating an account?
   c. How can you make your password secure?
2. Discuss common issues in form validation and how to handle them.
3. Explore additional validation options, such as server-side validation or more complex user feedback mechanisms.
4. Create a PowerPoint presentation to show your findings on additional validation options

Present your slides to your class for review and feedback.

DATABASE CONCEPTS

A **relational database** is a way of organising data into tables. This is just like spreadsheets, where information is stored in rows and columns. These tables are connected (or "related") using shared values, making it easy to manage and retrieve data.

Importance of a Database

A **well-designed relational database** is vital for an e-commerce website because it ensures smooth operations, accurate data, and a better customer experience. The following are some reasons why relational databases are important:

1. **Efficiently Manages Large Volumes of Data** - In an e-commerce system, you need to manage large amounts of data. This includes products, customers, and orders, which are all stored in separate tables that are connected. Each table is connected to the others through shared values, allowing the website to easily manage and update product listings, customer information, and orders, without duplicating data.

   For instance, when a customer places an order on Jumia, Melcom Ghana or Jiji, the database automatically links the customer with their order details, and updates the product stock levels.

2. **Ensures Data Integrity and Reduces Redundancy** - A well-designed relational database ensures that when a piece of information changes, like a customer's address, it updates across all relevant tables automatically, keeping everything accurate. Without a database, you might end up storing the same information in multiple places, creating unnecessary repetition, like people do in paper-based storage. With a relational database, the customer's information is stored once in the "Customers" table and linked to each order they place.

3. **Facilitates Complex Queries and Data Analysis** – Relational databases support complex queries that help the e-commerce business find specific products by name, category, or price range. Queries allow the e-commerce business to check a customer's purchase history or see how often a particular product is bought and they can generate sales reports for a specific period or by product category; allowing the business to analyse the data that it is storing.

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. Relational databases respond to requests in real-time (e.g., updating product stock or showing up-to-date customer info) and keep the site's operations smooth and quick even when multiple customers are interacting with the website simultaneously.

Activity 6.28

Simple Database Structure

1. Individually, create the following tables, using either paper or excel
   a. product
   b. customer
   c. employee
2. Discuss with your peers how these three tables are related to each other.

Key Components of a Relational Database

There are key components of databases that you must consider. These include the following:

1. **Tables (Relations)**

   A **table** is a collection of related data entries in a structured format consisting of **columns** and **rows**. Each table stores data about a specific entity. For example:

   a. Product Table: Holds details about items available for sale (e.g., product name, price, description).

   b. Customer Table: Stores information about customers who buy products (e.g., name, email, address).

   c. Order Table: Keeps records of transactions between customers and the store (e.g., order ID, order date).

   d. Transaction Table: Contains details about payments made by customers (e.g., payment method, transaction amount, etc).

Table 6.3 below shows an example of a product table

ProductID	ProductName	Price	Quantity
1	Wristwatch	GH¢ 373.30	4
2	Tecno Camon 30	GH¢ 3494.80	1
3	Toshiba Laptop	GH¢ 5536.00	1

2. **Columns (Fields/Attributes)**

   Each **column** in a table represents a specific **attribute** or property of the data being stored. For example, in our **Customers Table** (table 6.4), the columns include CustomerID, Name, and Email. These columns define the characteristics of the data stored in each row.

Table 6.4 below shows an example of a customer table

CustomerID	Name	Email
1	Daniel Miheso	miheso@example.com
2	Raphael Senyo Dordoe	dordoe@example.com
3	Mark Anibrika	anibrika@example.com

3. **Rows (Records/Tuples)**

   Each **row** (also called a **record** or **tuple**) represents an individual entry in the table. A row contains a set of values corresponding to each column in the table.

For example, in Table 6.4 above, each row represents one customer, with details like their Customer ID, name, and email.

4. **Entities and Attributes**
   a. **Entities** refer to the objects about which data is stored in the database. For instance, a product, a customer, or an order can all be considered entities in an e-commerce website.
   b. **Attributes** are the specific details or properties that describe an entity for example:
      i. **Product** - ProductID, Name, Description, Price, StockQuantity.
      ii. **Customer** - CustomerID, Name, Email, Address, PhoneNumber.
      iii. **Order** - OrderID, CustomerID, OrderDate, OrderTotal.
      iv. **Transaction** - TransactionID, OrderID, PaymentMethod, TransactionDate, Amount.

Relationships between Entities

Relationships link the different tables together using shared data. There are different types of relationships that can be established between tables, such as:

1. **One-to-Many (1:M):** This relationship means one entity in a table can be associated with many entities in another table. For example, a customer can place many orders, but each order is linked to only one customer.
2. **Many-to-Many (M:N**): This relationship means that multiple entities in one table can be associated with multiple entities in another table. For example, a product can appear in multiple orders, and an order can contain many products.

Activity 6.29

Designing Tables

1. Individually, on paper, design tables for customers, orders, products and transactions.
2. Define the relationships between these tables (e.g., one customer can place many orders).
3. Share your design with your peers in the class.

Keys in Databases

Primary Key

A **primary key** is a unique identifier for each record in a database table. It ensures that no two rows in a table have the same value for this field, to maintain order and integrity.

Key Characteristics

1. **Uniqueness** - Each value in the primary key column(s) must be unique. For instance, in a "Customers" table, CustomerID will be a primary key.
2. **Non-null** - A primary key cannot be empty. It means that a primary key cannot contain NULL (empty) values. This requirement ensures that every record has a valid and identifiable primary key value.
3. **Immutability** - The value of a primary key should remain unchanged throughout the record's lifecycle. Thus, the value should not change over time. If the value needs to change then the whole record needs to be deleted and a new record created, with a new unique identifier e.g. customerID.

Importance of Primary Keys

1. **Ensuring Data Integrity** — The primary key enforces uniqueness, preventing duplicate records from existing within the table and therefore maintaining the integrity of the data.
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 the tables. This referencing helps maintain referential integrity.

Foreign Key

A **foreign key** is a column in one table that references the primary key in another table. It creates a link between two tables, thereby enforcing a relationship.

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. This ensures that the value in the foreign key column exists in the primary key column of the referenced table. For instance, if a foreign key in the "Orders Table" refers to a "CustomerID" in the "Customers Table," the system will ensure that every "CustomerID" in the "Orders Table" exists in the "Customers Table."
2. **Nullable** - Foreign key columns can be empty (NULL). This means that some records may not need to be linked to another table. This allows flexibility in

relationships. For instance, a "Product Table" may include a column for OrderID as a foreign key. If a product has not yet been ordered, the OrderID column can remain NULL (empty).

3. **Consistency** – Foreign keys enforce consistency, by ensuring data in the related tables stay aligned and are valid. This prevents errors, like having an order that points to a deleted customer, or a payment linked to a non-existent order. If a customer is deleted from the "Customers Table", the database can either prevent the deletion or delete all related orders in the "Orders Table," thereby ensuring consistency.

Importance of Foreign Keys

1. **Establishing Relationships** - Foreign keys create relationships between tables, such as One-to-Many, where one customer can have many orders and Many-to-Many, where multiple orders can include multiple products. It connects data logically, enabling better organisation and retrieval. For instance, a "Customers Table" is related to an "Orders Table" by the foreign key, "CustomerID". This relationship shows which order(s) belong to which customer.
2. **Maintaining Data Integrity** - They ensure that references between tables are valid, avoiding issues like missing data and inconsistent data. This maintains trust in the database's accuracy. For instance, deleting a customer will ensure that all their orders are also deleted (cascading delete) or it prevents the deletion if orders still exist.
3. **Facilitating Joins** - Foreign keys make it easier to join tables during queries, allowing for complex data retrieval and analysis. For instance, an SQL query can join the "Customers Table" and "Orders Table" on the "CustomerID" foreign key to display customer names and their corresponding orders.

Examples

Consider an e-commerce database with the following tables

1. **Customer Table**: **In the customer table, the primary Key** is **CustomerID** (highlighted in red) as shown in **Figure 6.8**.

CustomerID	Name	Email
1	Daniel Miheso	miheso@example.com
2	Senyo Dordoe	*dordoe@example.com*
3	Mark Anibrika	anibrika@example.com

Figure 6.8: Customer Table

2. **Order Table:** In **Figure 6.9** below, the primary key is **OrderID** (highlighted in green) and the foreign key is **CustomerID** (referencing the **CustomerID** in the Customer table, highlighted in red). This relationship ensures that each order is associated with a valid customer.

OrderID	OrderDate	CustomerID
1011	2024-11-21	1
1012	2024-11-22	1
1013	2024-11-21	2

Figure 6.9: Order Table

3. **OrderProduct Table:** In **Figure 6.10** there are two primary keys (OrderID and ProductID). This is because the OrderID as a primary key is not unique since multiple products can be linked to the OrderID.

   For example, OrderID (1013) can be linked to products (B4 and B7).

   On the other hand, the ProductID can also be linked to more than one OrderID. Therefore, to get a unique key, we combine the two primary keys, which is known as a composite key (highlighted in blue).

OrderID	ProductID	Quantity
1013	B4	3
1012	C6	6
1013	B7	2

Figure 6.10: OrderProduct Table

Activity 6.30

Identifying Relationships

1. Create the following tables: Customers, Orders, Products
2. Add attributes to the tables.
3. Identify which fields should be primary or foreign keys and explain your reasons why.
4. Share your response with your peers in class.

Relational Database Tables and Fields

Tables and fields are essential components of a relational database, enabling structured storage and easy retrieval of information. For instance:

1. **Product Table** - Fields might include ProductID, Name, Description, Price and StockQuantity.

2. **Customer Table** - Fields might include CustomerID, Name, Email, Address and PhoneNumber.
3. **Order Table** - Fields might include OrderID, CustomerID, OrderDate, OrderTotal.
4. **Transaction Table** - Fields might include TransactionID, OrderID, PaymentMethod, TransactionDate, and Amount.

Schema Design

A **database schema** is a conceptual layout that defines the structure of the database. It outlines the tables, fields, relationships and constraints to ensure that the data remains accurate and accessible.

Table 6.5 Example of Schema Design of Product Table

Field	Type	Purpose	Constraints
ProductID	INT	Unique identifier for the product	PRIMARY KEY (ensures uniqueness)
Name	VARCHAR (255)	Product name	NOT NULL (every product must have a name)
Description	TEXT	Detailed product description	NULL allowed (optional field)
Price	DECIMAL (8, 3)	Product price	NOT NULL, CHECK (Price >= 0) ensures a valid price
StockQuantity	INT	Available stock	NOT NULL, CHECK (StockQuantity >= 0) ensures non-negative stock
Brand	VARCHAR(100)	Product brand name	NULL allowed (optional field)

Queries

SQL queries are written to perform various operations, such as retrieving, updating, or deleting data. This helps users to interact with a database to retrieve, update and manage the data.

Activity 6.31

Installation of MySQL

Caution: Ensure you are connected to the Internet for this activity.

1. Click on this link to download MySQL: *https://dev.mysql.com/downloads/installer/*
2. Click the download button of the second option (mysql-installer-community)
3. Select **No thanks, just start my download**

4. Wait for the download to complete
5. Double click the application to install it
6. Under the choosing a set up type, select **Full,** then click on next
7. Under check requirements, click on **execute** and wait for the status to upload
8. Click install the Microsoft Visual C++ if prompted
9. Click on next when the status column shows as "INSTALL DONE" for all items
10. On the installation popup page, click on execute to install the items
11. Click next three times repeatedly when done
12. Set a strong password that you can remember (leave this section for your teacher if you are using the school's computer)
13. Click next three times repeatedly when done
14. Click on Execute, wait for a while for it to complete
15. Click on finish
16. Click on next again
17. Click on finish
18. Click on next
19. Enter the password you set earlier, click on check and then click on finish
20. Click on next and finish again for a new window to open (Welcome to MySQL Workbench)

You are now ready to study and practice the basic operations that can be done on data in the database, using SQL queries.

Activity 6.32

SQL practice

1. Download this file by clicking on the link: *https://drive.google.com/file/d/1KF-dC14AGi5u3T66ntEhmqRcHYnAG6U0/view?usp=drive_link*
2. Watch this video tutorial and follow the instructions

   https://drive.google.com/file/d/1yG4UKdeQ1DQ-ojaC3zYhLg44-rDUjepE/view?usp=drive_link

SQL Statements

An SQL command (statement) is used to retrieve data from one or more table(s). It specifies the columns to be retrieved and the conditions for selecting records. This means it can filter, sort and group data by using additional clauses. For example:

1. **INSERT Statement**

   An SQL command is used to add new records to a table. It specifies the table and the values for each column. With one query, it can add multiple rows.

2. **UPDATE Statement**

   An 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.

3. **DELETE Statement**

   An SQL command is used to remove records from a table. It specifies the table and the conditions for deleting records. Be careful when using this statement, as it permanently deletes data from the database.

4. **WHERE Clause**

   An SQL clause which specifies the conditions that filter records in a query. It is used with SELECT, UPDATE, and DELETE statements. It can use multiple conditions with AND, OR and NOT (logic operators).

5. **ORDER BY Clause**

   An SQL clause that is used to sort data in ascending (default) or descending order. You can specify either ascending (ASC) or descending (DESC) order, if neither is specified then the data will be ordered in ascending order by default.

6. **GROUP BY Clause**

   The GROUP BY clause is used to group rows with the same values, in specified columns, into summary rows. It is commonly used with aggregate functions.

7. **JOIN Clause**

   The JOIN operation combines rows from two or more tables, based on a related column. There are different types of joins, including INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN.

8. **Aggregate Functions**

   Aggregate functions perform calculations on a set of values and return a single value. Common aggregate functions include COUNT(), SUM(), AVG(), MAX(), and MIN().

9. **Constraint**

   Constraints are rules applied to table columns, to ensure data integrity. Constraints include NOT NULL, UNIQUE, CHECK, and FOREIGN KEY.

Activity 6.33

SQL Statements

In groups of no more than five, complete the following

1. For each of the 9 SQL statements above, write down a real-world example of when that would be used in a database and the reason for using that statement
2. For each of the constraints, provide a definition and a reason why it might be important in a database:
   a. NOT NULL
   b. UNIQUE
   c. CHECK
   d. FOREIGN KEY
3. Share your responses with your peers

Activity 6.34

Organise yourselves into groups of no more than five. In your groups, using the e-commerce website that you created in **Activity 6.23** and the tutorials provided by your teacher, complete the following:

1. Create a database to hold and manage customer details (users of the website)
2. Add a feature for a new user to register for an online account, using a secure username and password
3. Add a feature to allow that new user to login
4. Add a feature to allow that new user to logout

EXTENDED READING

- Follow the link *here* or *https://www.w3schools.com/html/html_links.asp* to get more information on links in HTML.
- Refer to the URL *here* or *http://www.w3schools.com/html/html_media.asp* for more information on audio and video formats.
- The **DM ICT Centre** is an organisation that provides ICT Training and Education, Digital Literacy Programs, Software and Hardware Solutions, E-Learning Support, Internet Access and Digital Resources and Business and Entrepreneurship Support. The centre requires a website that can be designed and implemented by using HTML for structure, CSS for styling, and responsive frameworks like Bootstrap for its services. As a computing student, research and design this website that will best fit for DM ICT Centre.

Review Questions

1. List three key design elements you can add to a webpage using a web editor.
2. Why is UX (User Experience) important in web design.
3. What are the essential elements of an e-commerce web page?
4. What is a relational database model?
5. How would you add an image to a webpage using a web editor?
6. Why is it important to include a responsive design for an e-commerce webpage?
7. Write a query to retrieve all orders placed by a customer with CustomerID = 2.
8. How can you ensure secure transactions on an e-commerce website?
9. Why is accessibility an essential consideration in web design?
10. Analyse how integrating multimedia elements can improve the user interface of a webpage.
11. Compare the features of two popular e-commerce websites and identify key design elements that enhance user experience.
12. Given a relational database for your school's library system, propose a schema to include Books, Members, Loans, and Staff as best Computer Science student in the school.
13. Develop a relational database schema for Aflao Community Day Senior High School system.
14. Create a shopping product and cart tables for Jumia.
15. Create a personal web page which requests for personal information with placeholders, save and cancel buttons.

REFERENCES

1. Andrews, J., Dark, J., & West, J. (2019). CompTIA Network+ guide to networks (8th ed.). Cengage Learning.
2. Arya, N. (2024, June 15). AI/ML and deep learning: Types of machine learning. EJABLE. Retrieved November 18, 2024, from *https://www.ejable.com/tech-corner/ai-machine-learning-and-deep-learning/types-of-machine-learning/*
3. Barcodes Ghana. (2020). Guide to retail barcodes in Ghana. Barcodes Ghana. Retrieved June 20, 2024, from https://barcodesghana.com/wp-content/uploads/sites/96/2020/10/Guide-to-retail-barcodes-Ghana.pdf
4. Burdett, J., & Klenke, R. H. (2018). A survey of software-defined networking in network security applications. IEEE Transactions on Network and Service Management, 15(3), 1046-1061.
5. Chase, B. G., & Stone, A. R. (Eds.). (2021). *Foundations of computing systems: System software overview*. Pearson.
6. Chen, L. (2020, December 7). Logic and implementation of a spam filter: Machine learning algorithm. Retrieved November 18, 2024, from *https://towardsdatascience.com/logic-and-implementation-of-a-spam-filter-machine-learning-algorithm-a508fb9547bd*
7. Cisco Networking Academy. (2021). Introduction to networks. Cisco. Retrieved from https://www.netacad.com
8. Cisco Press. (n.d.). Network security basics. Retrieved October 31, 2024, from https://www.ciscopress.com/articles/article.asp?p=170742&seqNum=4
9. Cisco Systems. (2020). Cisco network design essentials. Cisco Press.
10. Cloudflare. (n.d.). What is a router? | Router definition. Retrieved October 31, 2024, from https://www.cloudflare.com/learning/network-layer/what-is-a-router/
11. Comer, D. E. (2018). Computer networks and internets (6th ed.). Pearson Education.
12. CompTIA. (2022). *CompTIA Network+ Certification Study Guide: Exam N10-008*. Downers Grove, IL: CompTIA Press.
13. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). The MIT Press
14. Forouzan, B. A. (2017). Data Communications and Networking (5th ed.). Boston, MA: McGraw-Hill Education.
15. Free Barcode. (n.d.). Manufacturer code of EAN-13 barcode. Retrieved 25 October 2024 from https://free-barcode.com/barcode/barcode-types/manufacturer-code-of-ean-13-barcode.asp
16. Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2014). *Data Structures and Algorithms in Python*. Wiley.

17. Google Cloud. (2023, March 11). Weather prediction with AI. Retrieved November 18, 2024, from *https://cloud.google.com/blog/topics/sustainability/weather-prediction-with-ai*

18. Grafiati. (n.d.). Network infrastructure. Retrieved October 31, 2024, from https://www.grafiati.com/en/literature-selections/network-infrastructure/

19. Gupta, N. (2024, April 15). A guide to supervised learning. Medium. Retrieved November 18, 2024, from *https://medium.com/@ngneha090/a-guide-to-supervised-learning-f2ddf1018ee0*

20. Javatpoint. (n.d.). Difference between LAN, MAN, and WAN. Retrieved October 31, 2024, from https://www.javatpoint.com/lan-vs-man-vs-wan

21. Kenji. (2024, March 24). Learn SQL basics in just 15 minutes. YouTube. https://www.youtube.com/watch?v=wmiDdBG-yP4

22. Khan Academy. (2021). Intro to relational databases. Retrieved from

23. *https://www.khanacademy.org/computing/computer-programming/sql* 22/11/2024.

24. Khan Academy. (2021). Intro to relational databases. Retrieved from *https://www.khanacademy.org/computing/computer-programming/html-css* 22/11/2024.

25. Khan, F. A., & Ghulam, A. (2020). "Internet Service Providers and Their Impact on Internet Accessibility." International Journal of Information and Education Technology, 10(5), 401-404

26. Kizza, J. M. (2020). Guide to computer network security (5th ed.). Springer.

27. Kozan, M. (2021, September 1). Supervised and unsupervised learning (an intuitive approach). Medium. Retrieved November 18, 2024, from *https://medium.com/@metehankozan/supervised-and-unsupervised-learning-an-intuitive-approach-cd8f8f64b644*

28. Krysik, A. (2024, July 2). Online shopping recommendations. Retrieved November 18, 2024, from *https://stratoflow.com/online-shopping-recommendations/*

29. Kurose, J. F., & Ross, K. W. (2017). Computer Networking: A Top-Down Approach (7th ed.). Boston, MA: Pearson.

30. Labels Online. (n.d.). ISBN barcode labels paper 40x25mm (BC1016). Labels Online. *https://www.labelsonline.co.uk/isbn-barcode-labels-paper-40x25mm-bc1016*

31. Lippman, D. (n.d.). Math in society: De Morgan's laws. LibreTexts. https://math.libretexts.org/Bookshelves/Applied_Mathematics/Math_in_Society_(Lippman)/17:_Logic/17.7:_De_Morgans_Laws

32. McCabe, J. D. (2013). Network analysis, architecture, and design (3rd ed.). Morgan Kaufmann.

33. Microsoft Corporation. (n.d.). *Configure network settings in Windows*. Microsoft Learn. Retrieved from *https://learn.microsoft.com/en-us/windows-server/networking/*

34. Microsoft. (2023, July 15). *What is application software?* Microsoft Learn. *https://learn.microsoft.com/en-us/windows/application-software-overview*

35. Odom, W. (2020). CCNA 200-301 official cert guide, volume 1. Cisco Press.

36. Olifer, N., & Olifer, V. (2005). Computer networks: Principles, technologies and protocols for network design. John Wiley & Sons.

37. OpenDNS. (n.d.). *DNS configuration for beginners*. OpenDNS. Retrieved from *https://www.opendns.com/setupguide/*

38. OpenStax. (n.d.). 4.2 If-else statements. In Introduction to Python programming. *https://openstax.org/books/introduction-python-programming/pages/4-2-if-else-statements*

39. Otosection. (2024, December 10). CSS grid responsive websites by using CSS grid system guide. Retrieved from *https://jethrojeff.com/*

40. Patel, A., & Patel, P. (2020). *Learn Data Structures and Algorithms with Golang*. Packt Publishing.

41. Pressman, R. S., & Maxim, B. R. (2019). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill.

42. Qodenext. (n.d.). What is UPC Barcode - Explained in detail. Qodenext. Retrieved 20 June 2024, from *https://www.qodenext.com/blog/what-is-upc/*

43. Quora. (n.d.). What are some examples of dial-up services? Retrieved November 1, 2024, from https://www.quora.com/What-are-some-examples-of-dial-up-services

44. Redicode. (n.d.). EAN-13 Barcode Images. Redicode. Retrieved 20 June 2024, from *https://redicode.com.au/product/ean-13-barcode-image/*

45. Reynolds, J., & Postel, J. (1996). RFC 1918: Address allocation for private internets. IETF. Retrieved from https://datatracker.ietf.org/doc/html/rfc1918

46. S B/Nasim. (2024, July 24). Broadband internet connection restored in Bangladesh. Risingbd. https://www.risingbd.com/english/national/news/105779

47. Schneider, J. B., Broschat, S. L., Dahmen, J. (2019). Algorithmic Problem Solving with Python. Page 22 -26.

48. Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating system concepts* (10th ed.). Wiley.

49. Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating system concepts* (10th ed.). Wiley.

50. Sommerville, I., & Sawyer, P. (2018). Software engineering and system software: A case study in their relationship. *Journal of Software Engineering Studies, 21*(3), 102-118. *https://doi.org/10.1016/j.softstud.2018.09.001*.

51. Stallings, W. (2017). Data and computer communications (10th ed.). Pearson.

52. Stallings, W. (2020). Foundations of modern networking: SDN, NFV, QoE, IoT, and cloud. Addison-Wesley Professional.

53. Stallings, W. (2021). System software: An overview. In B. G. Chase & A. R. Stone (Eds.), *Foundations of computing systems* (pp. 55-78). Pearson.

54. Tanenbaum, A. S., & Wetherall, D. J. (2010). Computer networks (5th ed.). Pearson Education.

55. Tanenbaum, A. S., & Wetherall, D. J. (2013). Computer Networks (5th ed.). Upper Saddle River, NJ: Pearson.

56. Tenenbaum, A. M., & Augenstein, M. J. (2015). *Data Structures Using C* (2nd ed.). Pearson.

57. Vidhya. (2024, February 6). Machine learning for social media. Retrieved November 18, 2024, from *https://www.analyticsvidhya.com/blog/2023/04/machine-learning-for-social-media/*

58. W3Schools. (2021). HTML tutorials. Retrieved from

59. W3Schools. (2021b). CSS tutorials. Retrieved from

60. Weiss, M. A. (2013). *Data Structures and Algorithm Analysis in Java* (3rd ed.). Pearson Education.

61. Wikipedia. (2023, October 31). Internet protocol suite. *https://en.wikipedia.org/wiki/Ip_stack*

62. Zhang, K., & Zhao, L. (2020). "Performance Evaluation of Internet Access Technologies." Journal of Communications and Networks, 22(2), 143-155.

GLOSSARY

A

Accessibility: The practice of designing websites, software, and interfaces to be usable by people with disabilities; ensuring inclusivity and equal access.

Account: A personal profile created on a system, website, or platform, often requiring login credentials, used to access personalised settings and services.

Artificial Intelligence (AI): The simulation of human intelligence in machines, enabling them to perform tasks such as learning, reasoning, and problem-solving.

B

Backtracking: An algorithmic technique for solving problems incrementally by trying out solutions and abandoning them if they fail to satisfy the problem's constraints.

C

Calculator: A software application used for performing mathematical calculations ranging from basic arithmetic to complex computations.

Carton Code: Is a barcode affixed to a carton to identify its contents, origin, and destination.

Case Sensitivity: The property of programming languages or systems where uppercase and lowercase letters are treated as distinct.

Chaining: A method of handling collisions in a hash table by storing all elements that hash to the same value in a linked list or chain.

Checkout: The process of completing a purchase in an online shopping system, typically involving payment and delivery information.

Collisions: This occurs when two data elements are assigned the same hash value.

Complex Data Types: Data types that combine multiple values, such as arrays, lists, dictionaries, and objects.

Compressing: The process of reducing the size of a file or data, such as images, videos, or documents, to save storage space or improve transmission speed.

Control Structures: Constructs in programming that dictate the flow of execution such as loops (for, while), conditional statements (if, else), and decision-making blocks.

D

Device drivers: Software that allows an operating system to communicate with hardware devices like printers, graphics cards, and keyboards enabling them to function properly.

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

Drag-and-Drop: A user interface interaction method where an item is clicked, dragged to a new location, and released, often used in file management or graphical design.

Dynamic Host Configuration Protocol: A network management protocol that automatically assigns IP addresses and other configuration settings to devices on a network enabling them to communicate effectively.

E

E-commerce: The buying and selling of goods and services over the internet.

Edges: In graph theory, an edge represents a connection or relationship between two nodes (vertices) in a graph.

Encryption: The process of converting data into a coded format to prevent unauthorised access, ensuring the confidentiality and integrity of the information.

Ethernet: A widely used technology for wired local area networks (LANs) that allows devices to communicate with each other using a common protocol.

F

File Transfer Protocol: A standard network protocol used to transfer files between a client and a server over a computer network.

File Transfer Protocol: A standard network protocol used to transfer files between a client and a server over a network such as the internet.

Firewall: A security system designed to monitor and control incoming and outgoing network traffic based on predetermined security rules.

Firmware: A type of software embedded in hardware devices that provides low-level control for the device's specific hardware.

First In First Out: A data handling principle where the first element added is the first one to be removed, it is used in structures like queues.

Function: A reusable block of code designed to perform a specific task, which can be called with arguments and may return a result.

G

Gateway: It is a payment gateway that facilitates online transactions.

Ghana Barcode System: Is a localised barcode system implemented in Ghana based on international standards to facilitate product identification, inventory management and trade.

Gigabit: A unit of data transfer speed equal to one billion bits per second (Gbps).

GUI (Graphical User Interface): A user interface that allows interaction with electronic devices through graphical elements like windows, icons, and buttons, rather than text-based commands.

H

Hypertext Transfer Protocol: The protocol used for transferring and displaying web pages on the internet. It defines how messages are formatted and transmitted between web browsers and servers.

I

Index: It is a reference point that allows for fast retrieval of data.

Internet: A global network of interconnected computers and devices that allows the sharing of information and communication worldwide.

Inverted index: A data structure used in search engines to map content, such as words, to their locations in a document or set of documents, enabling efficient full-text searches.

L

Labelled data: Data that has been annotated with information, such as tags or categories, often used in supervised learning models in AI.

Last In First Out: A data handling principle where the last element added is the first one to be removed, it is used in structures like stacks.

O

Online Shopping Recommendations: AI-driven systems that analyse user behaviour, preferences, and data to suggest products during online shopping.

Open addressing: A collision resolution strategy in hash tables where upon collision, alternative slots are searched within the table itself to store the element.

Operating System: Is type of a system software that manages hardware resources and provides common services for computer programs.

P

Phone Book: A common analogy used in computing to explain data structures like hash tables or databases, where names (keys) are associated with phone numbers (values).

Primitive Data Types: Basic data types provided by a programming language, such as integers, floats, characters, and Booleans.

Processor: The central processing unit (CPU) of a computer.

Productivity software: Applications designed to help users perform tasks efficiently.

Program: A sequence of instructions written in a programming language that a computer can execute to perform a specific task.

Programming Concepts: Fundamental ideas or principles used in writing and understanding computer programs such as variables, loops, functions, and algorithms.

Programming language: A formal set of rules and syntax used to communicate instructions to a computer. Examples include Python, Java, and C++.

R

Real-Time: Systems that provide immediate feedback or updates without noticeable delay. Examples include live streaming and instant messaging.

Relation: A logical connection between data entities in databases.

Reserved Keywords: Words that have predefined meanings in a programming language and cannot be used as identifiers such as if, else, for, and return.

S

Search engines: Software systems designed to search and retrieve information from the internet or a database, based on user queries such as Google, Bing, or DuckDuckGo.

Secure Socket Layer: A predecessor to TLS, SSL is a cryptographic protocol that established secure connections over the internet commonly used for securing websites before being replaced by TLS.

Shopping: The act of browsing and purchasing items, which can now also be done online through e-commerce platforms.

Siblings: In tree structures, siblings are nodes that share the same parent node.

Simulation tools: Software applications used to model and mimic real-world processes or systems for analysis, education, or testing. Examples include MATLAB and Simulink.

Special Characters: Non-alphanumeric characters that have specific meanings or uses in programming, such as @, #, $, *, and &.

Sub-tree: A smaller tree within a larger tree structure, consisting of a node and all its descendants.

T

Transport Layer Security: A cryptographic protocol that provides secure communication over a computer network ensuring data confidentiality, integrity and authentication.

U

Universal Product Code: Is a barcode used worldwide for tracking trade items in stores.

User: An individual who operates or interacts with a computer system, software, or application to achieve a specific purpose.

Utilities: Is a type of System software designed to perform specific tasks to maintain, optimise, or protect a computer system such as antivirus programs, disk cleanup tools or backup utilities.

W

Web browsers: Software applications that enable users to access, retrieve, and view content on the World Wide Web.

Web: Short for the World Wide Web, it is a system of interconnected documents and multimedia accessible over the internet using web browsers.

Websites: Collections of interconnected web pages hosted on a server, forming a single entity with a shared domain name.

Wi-Fi Protected Access: A security protocol developed to secure wireless networks by encrypting data and authenticating users. It replaced the less secure WEP (Wired Equivalent Privacy) standard.