

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

(In the Name of Allah, the Most Compassionate, the Most Merciful.)

COMPUTER SCIENCE AND ENTREPRENEURSHIP



Based on Revised National Curriculum of Pakistan 2023



PUNJAB EDUCATION, CURRICULUM,
TRAINING AND ASSESSMENT AUTHORITY

Authors

- **Prof. Dr. Muhammad Atif**

Dean faculty of C.S.
Lahore Garrison University Lahore.

- **Prof. Dr. Syed Waqar ul Qounain Jaffry**

Chairman Dept. of IT, University of The Punjab, Allama Iqbal
Campus (Old Campus) Shahrah-e-Quaid-e-Azam, Lahore.

External Review Committee

- **Prof. Dr. Arshad Ali**

FAST School of Computing, National
University of Computer and Emerging
Sciences (NUCES), Lahore Campus.

- **Dr. Mudasser Naseer**

Associate Prof. (CS), Department of CS
& IT, University of Lahore, Lahore.

- **Mrs. Tabinda Muqaddas**

Assistant Prof. (CS), Govt.
Associate College for Women,
Gulshan Ravi, Lahore.

- **Prof. Dr. Shahzad Asif**

Chairman Department of CS, UET,
Lahore.

- **Dr. Nadeem Iqbal**

Associate Prof.(CS), Department of CS &
IT, University of Lahore, Lahore.

- **Dr. Abdul Sattar**

Assistant Professor (CS),
Lahore Garrison University.

- **Dr. Adeel Nisar**

Assistant Prof. (CS), University of The
Punjab, Lahore.

- **Mr. Fahad Asif**

EST (CS), Govt. Lab Higher Secondary
School, PECTAA Kasur.

Reviewers of SED

- **Kashif Shahzad Ch**

SST(CS/IT), Govt. Muslim High School
No. 2, Civil Lines, Lahore.

- **Ejaz Hussain**

SST(CS/IT), Govt. Boys High School,
Gawalmajdi, Lahore(Ex-MCL)

- **Samina Khaliq Munir**

SST(CS), Govt. Muhammadiyah
Girls High School, Data Nagar,
Badami Bagh, Lahore.

- **Ansar Raza**

SST(CS/IT), Govt. Millat High School,
Mughalpura, Lahore.

- **Salma**

SST(IT), Govt. Model Girls High School,
Samanabad, Lahore.

- **M.Naveed ur Rehman**

SST(CS/IT),Govt. High School
Chung Multan Road Lahore.

- **Shabana Almas**

SST(IT), Govt. Pilot Secondary School for
Girls, Wahdat Colony, Lahore.

- **Maryam Ashraf**

SST(CS), BGHS School, Dev Samaj
Krishan Nagar, Lahore.

- **Ch. Ghazanfar Hussain**

SST(CS/IT), Govt. Iqbal High
School, Garhi Shahu Lahore.

Reviewers of HED

- **Abdul Wahab**

Asst. Prof. (CS), Govt. Graduate College
of Science Wahdat Road, Lahore.

- **M. Zeeshan Shabbir**

Asst. Prof. (CS), Govt. Graduate College
of Science Wahdat Road, Lahore.

- **Fraz Shoukat**

Lecturer (CS), Govt. M.A.O
Graduate College, Lahore.

Supervision

Jahanzaib Khan, Assistant Director (C.Sci.), PECTAA

Deputy Director (Compliance-Science)

Syed Saghir-ul-Hassnain Tirmizi

Director (Curriculum & Compliance)

Aamir Riaz

Incharge Art Cell

Aisha Sadiq

Layout & Designing

Tahir Sarfraz Butt

Illustration

Ayat Ullah, Tahir Sarfraz Butt

< This Textbook is based on Revised National Curriculum of Pakistan 2023. >

All rights are reserved with the Punjab Education, Curriculum, Training and Assessment Authority.

No part of this textbook can be copied, translated, reproduced or used for preparation of test papers, guidebooks, keynotes and helping books.

CONTENTS

Unit	Topic	Page
1	Introduction to Computational Systems	1
2	System Design and Troubleshooting	25
3	Introduction to Computer Networks	47
4	Computational Thinking	68
5	Web Development with HTML, CSS and JavaScript	91
6	Data Science and Data Gathering	114
7	Emerging Technologies in Computer Science	128
8	Ethical, Social, and Legal Concerns in Computer Usage	139
9	Entrepreneurship in Digital Age	148
	Answers	163
	Instructions for Preparation of Exam Paper	164
	Model Paper	166



UNIT 1

Introduction to Computational Systems



Student Learning Outcomes

By the end of this chapter, students will be able to:

- Define and describe general system theory, its types, objectives, components.
- Explain the importance of system objectives such as processing information, supporting applications, and achieving specific goals.
- Describe a computer as a system, including its objectives, architecture, components, and interactions.
- Recognize the role and importance of computer system components and their interactions.
- Understand the Von Neumann architecture and its core components: CPU, motherboard, memory, storage devices, input/output ports, and devices.
- Explain the relationship between the CPU, memory, and storage, and how data flows within a system.
- Describe how components within a computer system interact to execute tasks, such as how the CPU fetches, decodes, and executes instructions stored in memory.
- The different number systems, including decimal, binary, hexadecimal, and octal, and their respective base values and digits.
- The representation of whole and real numbers in a computer, including binary encoding methods for both.
- How various arithmetic operations, such as addition, subtraction, multiplication, and division, are performed on binary representations of numbers?
- The concept of common text encoding schemes, such as ASCII and Unicode, and how they represent characters.
- Key terms related to data representation, including ASCII, Unicode, binary, signed and unsigned numbers, bits, bytes, hexadecimal number systems, negatives in binary, two's complement and binary arithmetic
- Identify and explain the significance of system software and application software.
- Understand the role and main functions of system software.
- Demonstrate and differentiate between system software and application software, understanding their roles within a computer system.



Introduction

This chapter provides an overview of the theory of system, introduces fundamental concept and explores various types of systems. It begins by defining what a system is, discusses its basic component, objectives, environments and methods of communication. Then it describes Von Neumann Architecture and its working. We will explore the significance of system software and application software. We will delve into various number systems, their applications, their interconversion and how they are used in computers.

1.1 Theory of Systems

A **system** is an organized collection of **interdependent parts** that work together to perform a specific **function** or achieve a common **goal**. The idea of a system is useful to explain both the external reality as well as the internal one. An **Information System** is a combination of technology and people that collects, processes, stores, and shares data to help an organization make decisions and operate effectively. An Information System is simply an organized set of components that are coordinated to perform a designated function. All the components of the system are in some way related to each other and the functioning of the other components enhances the operation of the system.

Let us consider a simple example, such as a Computer System, depicted in **Figure 1.1**. every computer, machine, or device has components that work together to achieve an outcome.



Fig 1.1 Computer System



Basic Concepts of Systems

A system is described by its objectives, components, communication among components and environment in which it works. The components of a system communicate with each other to achieve the system's objective in an environment. Systems can be simple, like a thermostat, or complex, like the human body or a computer network.

Objective

Every system has a purpose or goal that it wishes to fulfil. Analyzing a system's operation requires understanding its aim. This insight improves the efficiency and efficacy of the present system.

A transport system aims to transfer people and products securely and effectively between locations. A computer system's principal goal is to process data and provide useful information to users.

Components

Components are the building blocks of any system. Each component plays a specific role and contributes to the overall functionality of the system. Understanding the role of each component of the system is essential to understand how the entire system works. This helps in identifying problems, improving performance, and refining system design. Smooth and proper working of these components together ensures the system meets its objectives.

Environment

The environment of a system includes everything external to the system that interacts with it. It consists of all external factors that affect the system's operation. Understanding the environment of a system is important as it influences the system's performance and behavior by providing inputs and receiving outputs. Intelligent systems adjust to changes in their environment to continue their functionality.

Communication

Communication and interaction among system components is key to the functioning of a system. It ensures that components work together in an organized and smooth manner to achieve the system's objectives. For example, in a computing system the CPU communicates with memory to fetch and store data, and in a biological system brain sends signals to muscles to initiate movement.



Class ACTiViTY



Design a Simple System

Objective: To apply the principles of system design and understand the process of creating a functional system.

Required Material: Computers or tablets with diagramming software (e.g., Lucidchart), paper, pencils, markers.

Activity Type: Pair

Activity Tasks Detail: Begin with an introduction where the teacher presents an example of a simple system. Students will then work in pairs to define the objective of their chosen system, list its components, describe their interactions, and outline the system's environment. The pairs will use diagramming software to create a system prototype or diagram. Finally, they will present their designs to the class in a review and feedback session.

Output: Each pair will produce a system prototype or diagram and receive feedback to refine their design ideas.

1.2 Software

Software is a collection of programs and instructions that tell a computer what to do and how to do. Without software, computers would be useless machines.

Did You Know?

The first computer virus, called "Creaper," was created in 1971 as an experimental self-replicating program. It simply displayed the message, "I'm the creeper, catch me if you can!"

Types of Software

System Software

System software is designed to manage the system resources and provide a platform for application software to run. It acts as a bridge between the hardware and the user applications. Here are some examples:

- **Operating Systems:** Examples include Microsoft Windows, macOS, and Linux.
- **Device Drivers:** These include printer drivers, graphics card drivers, and sound card drivers.
- **Utility Programs:** Examples are antivirus software, disk cleanup tools, and backup software.

Application Software

Application software is designed to help users perform specific tasks. These programs are built to fulfill user needs and are typically more varied than system software. Examples include:



- **Word Processors:** Such as Microsoft Word and Google Docs.
- **Web Browsers:** Such as Google Chrome, Mozilla Firefox, and Safari.
- **Games:** Such as Minecraft, Fortnite, and Among Us.
- **Media Players:** Such as VLC Media Player and Windows Media Player.

Differentiating Between System Software and Application Software

- **Purpose:** System software manages and operates computer hardware, making it possible for application software to run. Application software helps the user to perform specific tasks.
- **Examples:** System software includes operating systems and device drivers. Application software includes word processors, web browsers, and games.
- **Installation:** System software is usually pre-installed on a computer, while application software can be installed by the user as needed.



TidBiTs

Always keep your system software updated to ensure your computer runs smoothly and is protected from security threats.

Class ACTiViTY



Make a list of all the software you use on your computer or tablet. Categorize them into system software and application software. Discuss with your classmates which software you find most useful and why.

1.3 The Architecture of von Neumann Computers

The Von Neumann architecture is a computer paradigm that describes a system in which the hardware of the computer has four primary components: the memory, the Central Processing Unit (CPU), input mechanisms, and output mechanisms. This model is called the John von Neumann model, the Neumann model named in honor of the mathematician and physicist who contributed to its development during the 1940s.

Components

Now we will look at brief overview of the key parts that constitute the architecture of the von Neumann computer.



1. **Memory:** Contains both input data and the instructions (program) required for CPU processing. For instance, consider the RAM of your computer: when a program starts it is loaded into RAM to enable faster execution compared to when it runs from the hard disk.
2. **Central Processing Unit (CPU):** Performs addition and subtraction, and executes commands provided by the memory. The system has two main components: the Arithmetic Logic Unit (ALU) and the Control Unit (CU).
The Arithmetic Logic Unit (ALU) performs mathematical computations and logical operations.
A **Control Unit (CU)** is a peripheral that governs the activities of the CPU by instructing the ALU and memory to execute tasks according to the program instructions. It ensures the proper and timely execution of duties by all the other components.
When doing the calculation $2 + 2$ on a calculator application, the Arithmetic Logic Unit (ALU) handles the numerical values while the control Unit (CU) supervises the whole procedure.
3. **Input Devices:** Enable users to input data and instructions into the computer system.
Illustrative examples include keyboard, mouse, and microphone. Entering text on the keyboard transmits data to the CPU for subsequent processing.
4. **Output Devices:** Present or communicate the outcomes of the tasks executed by the computer. Consider, for instance, a monitor and printer. Upon completion of data processing, the CPU transmits the outcome to the monitor for visual display.

A **System Bus** is a communication mechanism that facilitates the movement of data between components inside a computational system. It comprises:

Data Bus: Transports data.

Address Bus: Maintains data destination information.

Control Bus: Transports control electrical signals.

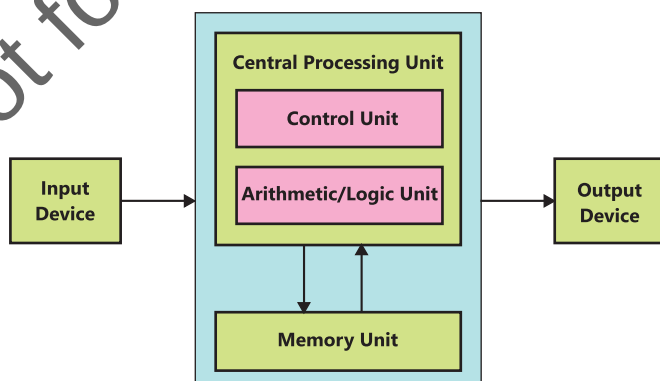


Fig. 1.2 Von Neumann computer Architecture



Working of Von Neumann Architecture

The Von Neumann architecture encompasses four essential stages for a CPU to carry out instructions, namely fetching, decoding, execution, and storage. To demonstrate this procedure, we will use the example of addition of two-digits with a basic calculator application.

- **Fetching:** The central processing unit retrieves an instruction from the computer's memory. This instruction specifies the operation to be executed by the CPU.

Comprising Components: Memory, CPU, Program Counter (PC), Instruction Register (IR)).

Description: The Program Counter (PC) stores the memory address of the subsequent instruction. Once the address is stored in memory, the instruction located at that location is retrieved and placed into the Instruction Register (IR).

- **Decoding:** In order to determine the necessary action, the Control Unit (CU) decodes the instruction.

Comprising Components: Control Unit (CU).

Description: The control unit (CU) decodes the opcode (operation code) of the instruction and determines the required procedures and data.

- **Execution:** The CPU processes the instruction. When the instruction involves a computation, it is executed by the Arithmetic Logic Unit (ALU). Any task that requires transferring data between several locations is managed by the CU.

Comprising Components: ALU, CU.

Description: The Arithmetic and Logic Unit (ALU) carries out mathematical and logical calculations, while the Control Unit (CU) handles data transmission activities.

- **Storing:** The outcome of the computation is either returned to memory or sent to an output device.

Comprising Components: Memory and Output Device.

Description: The outcome is either stored in a designated memory location or sent to an output device, such as a display.



1.4 Number Systems

Number systems are essential in computing because they form the basis for representing, storing, and processing data. Different number systems help computers perform tasks like calculations, data storage, and data transfer. These systems allow computers to represent various kinds of information, such as text, colors, and memory locations. Here is a description of a few number systems.

Decimal Number System

The decimal number system is a base-10 number system that consists of digit from 0 to 9 and we use it in everyday life. That's why each digit of the number represents a power of 10. In the decimal system the place values starting from the rightmost digits are 10^0 , 10^1 , 10^2 , and so on. For example, the decimal number 523 means:

$$5 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 = 500 + 20 + 3 = 523$$

Binary Number System

In binary, the place values are arranged from the right to left, starting with 2^0 , and ending at 2^n , where each position represents a power of 2. For example, the binary number 1011 can be converted to decimal as follows:

$$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 0 + 2 + 1 = 11_{10}$$

Computers work in binary system especially because this method fits well with electronics. Digital circuits have two states: They can be either on or off. These states are easily represented by the binary digits: 1 represent ON, and 0 represents OFF. When typing on the keyboard, the computer translates every letter to a binary. Similarly, number, text, images, and sound are all, at their lowest level, reduced to binary. We shall discuss this in details later in this chapter.

When you type a letter on your keyboard, the computer converts it into a binary code. Similarly, all types of data, including numbers, text, images, and sounds, are ultimately broken down into binary code. We will explore it further later in this chapter.

Conversion from Decimal to Binary

The following algorithm converts a decimal number to binary.

1. To convert decimal number to binary form, divide the decimal number by 2.
2. Record the remainder.
3. Divide the number by 2 until the quotient which is left after division is 0.
4. Meaning it is represented by the remainders and it's read from the bottom to the top for the binary number.



Example: Convert 83 to binary

$$83 / 2 = 41 \text{ remainder } 1$$

$$41 / 2 = 20 \text{ remainder } 1$$

$$20 / 2 = 10 \text{ remainder } 0$$

$$10 / 2 = 5 \text{ remainder } 0$$

$$5 / 2 = 2 \text{ remainder } 1$$

$$2 / 2 = 1 \text{ remainder } 0$$

$$1 / 2 = 0 \text{ remainder } 1$$

2	83
2	41—1
2	20—1
2	10—0
2	5—0
2	2—1
2	1—0
	0—1



Figure 1.3 Decimal to Binary conversion

The above steps are graphically shown in Figure 1.3. If the remainders are read from bottom to top then it gives the required result in binary, which is 1010011.

Octal Number System

Octal is a positional numeral system with base eight, which implies that a digit to be used ranges from 0 to 7. The last digit is a single digit power of 8 while the other digits are the coefficients. In the octal system, the place values starting from the 8^0 , 8^1 , 8^2 and so on. For example, the octal number 157 means, $1 \times 8^2 + 5 \times 8^1 + 7 \times 8^0 = 64 + 40 + 7 = 111_{10}$.

Conversion from Octal to Binary

Each octal digit represents three binary digits (bits) because the octal system is base-8, and the binary system is base-2. This relationship arises from the fact that 8 is a power of 2 ($8 = 2^3$). So, each octal digit can be precisely represented by three binary digits (bits). This means that any value from 0 to 7 in octal can be converted into a 3-bit binary number. This relationship makes conversion between binary and octal straight forward. Table 1.1 shows the correspondence between octal and binary digits:

Example:

Consider the 9-bit binary number 110101011. This number can be divided into groups of three

Bits from right to left:

110 101 011

Each group of three bits corresponds to a single octal digit:

$$110 = 6$$

$$101 = 5$$

$$011 = 3$$

Octal	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Table 1.1: Correspondence between Octal and Binary Digits



So, the binary number 110101011 is equal to 653 in octal.

Note that the octal number system isn't actually used in modern computers to do their work. Therefore, we can say that the binary number 110101011 is equal to 653 in octal. Whenever you have a binary number that cannot be divided into groups of a three, you'll have to add zero up to the left end of it to make it appropriate.

Conversion from Decimal to Octal

The algorithm below converts a decimal number into an octal.

1. To convert the decimal number to an equivalent octal number, divide the number by 8.
2. Write down the remainder.
3. After that divide the obtained quotient by 8.
4. Continue the divisions until one of the numbers results in 0.
5. Octal is a base eight number and the octal number is the remainder read from the bottom up to the top.

8	83	
8	10	— 3
8	1	— 2
	0	— 1

Figure 1.4: Conversion from Decimal to Octal.

Example: Convert 83 to octal

- $83 / 8 = 10$ remainder 3
- $10 / 8 = 1$ remainder 2
- $1 / 8 = 0$ remainder 1

The above steps are graphically shown in Figure 1.4. Going up from bottom, the remainder reading will give the desired result, that is 123 in the octal system.

Class ACTIVITY



1. Work in pairs to convert the following decimal numbers to octal: 45, 128, 64.
2. Convert these octal numbers to decimal: 57, 124, 301.
3. Share your answers with the class and discuss any differences.

Did You Know?

The octal system was used in early computing systems like PDP-8. It was used because it is easier to convert between octal and binary than between decimal and binary.



Hexadecimal Number System

The hexadecimal is a base 16 number system with digit number from 0 to 9 and alphabets from A to F; each digit represents the power of 16 to show the position of the digit. The letter A to F stand for the numeric value of 10 to 15, The digits in hexadecimal move from right to left in place value that are 16^0 , 16^1 , 16^2 ... anothers. For example, the hexadecimal number 1A3 can be represented in decimal as:

$$1 \times 16^2 + A \times 16^1 + 3 \times 16^0 = 1 \times 256 + 10 \times 16 + 3 \times 1 = 256 + 160 + 3 = 419_{10}$$

Conversion from Hexadecimal to Binary

The hexadecimal number system is not directly used by computers either. However, it provides an even more compact representation than octal. This makes it easier for us to read and write large binary numbers.

This is because the hexadecimal system is base-16 and the binary system is base-2, therefore every single hexadecimal digit equals four binary bits. This relationship stems from the fact that 16 is a power of 2 ($16 = 2^4$). This means that any hexadecimal number between 0 and 15 can be converted into 4-bit binary number.

Table 1. 2 illustrates conversion of hexadecimal to binary digits. Each group of four bits corresponds to a single hexadecimal digit.

Example:

Therefore, the binary number 1101011010110010 equals to the hexadecimal number D6B2. In case a binary number cannot be grouped as four bits add zero(s) to the left of the number to make it fit.

1101 0110 1011 0010

Hexadecimal	Binary	Hexadecimal	Binary
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

Table 1.2: Correspondence between Hexadecimal and Binary Digits



$$1101 = D$$

$$0110 = 6$$

$$1011 = B$$

$$0010 = 2$$

Converting Decimal to Hexadecimal

The following algorithm converts a decimal number to hexadecimal:

1. Convert the decimal number to an absolute value by dividing it by 16.
2. Record the quotient and the remainder.
3. Continue dividing the quotient by 16 and write down the remainder until the quotient is zero.
4. The hexadecimal number, as you might have guessed, is the remainder read from bottom to top.

Example: Convert 2297 to hexadecimal

$$2297 / 16 = 143 \text{ remainder } 9$$

$$143 / 16 = 8 \text{ remainder } F$$

$$8 / 16 = 0 \text{ remainder } 8$$

16		2297	
16		143 — 9	⇒ 9
16		8 — 15	⇒ F
		0 — 8	⇒ 8 ↑

The above steps are graphically shown in Figure 1.5.

Reading the remainders from bottom to top gives the required result, i.e., 8F9 in hexadecimal.

Figure 1.5: Decimal to Hexadecimal

Class Activity

Find the following values and express them in hexadecimal. Discuss your findings with your classmates:

- Minimum Age to Cast Vote
- Length of the Indus River
- Total Districts in Pakistan
- Height of K2 (the second-highest mountain in the world)
- Area of Pakistan

1.5 Data Representation in Computing Systems

Computers can process and store a lot of information. In the following section we will discuss numeric data representation.



Whole Numbers (W)

Whole numbers are a set of non-negative integers. They include zero and all the positive integers. Mathematically, the set of whole numbers is:

$$W = \{0, 1, 2, 3, \dots\}$$

In computing, whole numbers are often used to represent quantities that can't be negative. Examples include the number of students in a school, a person's age in years, and grades, provided there are no negative figures such as credit point balances.

A 1-byte integer has 8 bits to store values. If all 8 bits are on, it represents the maximum value, 11111111_2 , which is 255_{10} . If all bits are off, it represents the minimum value, 00000000_2 , which is 0_{10} . Similarly, using 2 or 4 bytes, we get more bits to store data allowing us to store bigger values. If n is the number of bits, the maximum value that can be represented is $2^n - 1$ for examples:

- 1-Byte whole number (8 bits): Maximum value = $2^8 - 1 = 255$
- 2-Byte whole number (16 bits): Maximum value = $2^{16} - 1 = 65,535$
- 4-Byte whole number (32 bits): Maximum value = $2^{32} - 1 = 4,294,967,295$

Integers (Z)

Integers extend the concept of whole numbers to include negative numbers. In computer programming, we call them signed integers. The set of integers is represented as:

$$Z = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$$

To store both positive and negative values, one bit is reserved as the sign bit (the most significant bit). If the sign bit is ON(1), the value is negative; otherwise, it is positive. Using this system, the maximum positive value that can be stored in a 1 byte signed integer is $(01111111)_2$, which is 127_{10} . As the bits available to store a value is $n - 1$, hence the maximum value will be $2^{n-1} - 1$. We can use this formula to compute the maximum values for 2 and 4 bytes.

Negative values are stored using 2's complement, explained in the following section.

1's Complement

1's Complement is a binary number representation system used to express positive and negative numbers in digital systems. It's one of several methods for representing signed numbers in binary. 1's Complement represents negative numbers in binary by inverting all the bits (changing 0s to 1s and 1s to 0s).

Negative Values and 2's Complement

To store negative values, computers use a method called 2's complement. To find the 2's complement of a binary number, follow these steps:



1. Invert all the bits (change 0s to 1s and 1s to 0s). (1's complement)
2. Add 1 to the Least Significant Bit (LSB).

Example: Let's convert the decimal number -5 to an 8-bit binary number:

1. Start with the binary representation of 5: 00000101_2 .
 2. Invert all the bits: 11111010_2 .
 3. Add 1: $11111010_2 + 1_2 = 11111011_2$.
- So, -5 in 8-bit 2's complement is 11111011_2 .

Minimum Integer Value

For an 8-bit integer, we switch on the sign bit for the negative value and turn all bits ON. resulting in 11111111_2 . Except the first bit, we take two's complement and get 10000000_2 which is 128_{10} . Thus minimum value in 1-byte signed integer is -128, i.e., -2^7 . The minimum value is computed using the formula -2^{n-1} , where n is the total number of bits.

- **2-Byte Integer (16 bits):** Minimum value = $-2^{15} = -32,768$
- **4-Byte Integer (32 bits):** Minimum value = $-2^{31} = -2,147,483,648$

1.6 Binary Arithmetic Operations

Arithmetic operations include addition, subtraction, multiplication and division, which are performed on two numbers at a time. Binary arithmetic operations are similar to decimal operations but follow binary rules. Here's a brief overview of the basic operations:

Addition

Binary addition uses only two digits: 0 and 1. Here, we will learn how to add binary numbers and how to handle the addition of negative binary numbers.

Binary Addition Rules

Binary addition follows these simple rules:

- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 0$ (with a carry of 1 to the next higher bit)

Example of Binary Addition

**Example 1:**

$$\begin{array}{r}
 1101 \\
 +1011 \\
 \hline
 11000 \\
 \hline
 \end{array}$$

In this example:

- $1 + 1 = 0$ (carry 1)
- $0 + 1 + 1$ (carry) = 0 (carry 1)
- $1 + 0 + 1$ (carry) = 0 (carry 1)
- $1 + 1 + 1$ (carry) = 1 (carry 1)

Subtraction

In binary arithmetic, subtraction can also be carried out by adding the 2's complement of the subtrahend to the minuend.

Example: Subtract 6 from 9 in Binary

$$\text{Minuend} = 9_{10} = 1001_2$$

$$\text{Subtrahend} = 6_{10} = 0110_2$$

Step 1: Find the Two's Complement of the Subtrahend

- Invert the bits of 0110_2 :

$$\text{Inversion: } 1001_2$$

- Add 1 to the inverted number:

$$1001_2 + 1_2 = 1010_2 = -6_{10}$$

Step 2: Add the Minuend and the Two's Complement of the Subtrahend

$$1001_2 + 1010_2 = 10011_2$$

Step 3: Discard the Carry Bit

$$10011_2 \quad \text{Discard carry} \quad 0011_2 = 3_{10}$$

$$\text{So, } 9 - 6 = 3.$$

Multiplication

Binary numbers are base-2 numbers, consisting of only 0s and 1s. Multiplying binary numbers follows similar principles to multiplying decimal numbers, but with simpler rules. Here, we will learn how to multiply binary numbers with example.

Steps to Multiply Binary Numbers

1. Write down the binary numbers, aligning them by the least significant bit (rightmost bit).
2. Multiply each bit of the second number by each bit of the first number, similar to the long multiplication method in decimal.



3. Shift the partial results one place to the left for each new row, starting from the second row.
4. Add all the partial results to get the final product.

Example

Let's multiply two binary numbers: 101_2 and 11_2

$$\begin{array}{r}
 101 \\
 \times 11 \\
 \hline
 101 \\
 101 \\
 \hline
 1111
 \end{array}$$

(This is $101_2 \times 1_2$)
 (This is $101_2 \times 1_2$, shifted left)

So, $101_2 \times 11_2 = 1111_2$

Did You Know?

The Central Processing Unit (CPU) of a computer performs millions of binary multiplications every second to execute complex instructions and run programs!

Division

Binary division is similar to decimal division but only involves two digits: 0 and 1. It follows steps like comparing, subtracting, and shifting, similarly to long division in the decimal system.

Steps of Binary Division

1. **Compare:** Compare the divisor with the current portion of the dividend.
2. **Subtract:** Subtract the divisor from the dividend portion if the divisor is less than or equal to the dividend.
3. **Shift:** Shift the next binary digit from the dividend down to the remainder.
4. **Repeat:** Repeat the process until all digits of the dividend have been used.

Example

Divide 1100_2 by 10_2

$$\begin{array}{r}
 10 \overline{) 1100} \quad 110 \\
 \underline{-10} \\
 10 \\
 \underline{-10} \\
 0
 \end{array}$$

(Step 1: Compare 10 with first two 11, subtract 10 from 11)

(Step 2: Bring down the next digit 0)

(Step 3: Compare 10 with 10, subtract 10 from 10)

(Step 4: Bring down the next digit 0, no more digits left)

Result: $1100_2 / 10_2 = 110_2$



Class ACTiViTY

Practicing Binary Division

Objective: To practice and understand binary division through hands-on examples.

Instructions:

1. Form groups of three to four students.
2. Each group will solve the following binary division problems:
 - (a) $10101_2 \div 10_2$
 - (b) $11100_2 \div 11_2$
 - (c) $100110_2 \div 101_2$
3. Write down each step of your division process clearly.
4. Present your solutions to the class, explaining each step and the reasoning behind it.

1.7 Common Text Encoding Schemes

Text encoding schemes are essential for representing characters from various languages and symbols in a format that computers can understand and process. Here are some of the most common text encoding schemes used in computers:

ASCII

ASCII is an acronym that stands for American Standard Code for Information Interchange. It is a character encoding standard adopted for representing in devices such as computers and similar systems that use text. Each alphabet, number or symbol is given a code number between 0 and 127 as shown in Table 1.3.

ASCII enables different computers and devices to exchange text information reliably. Let's encode the name of our country (Pakistan) using ASCII.

- The ASCII code for an upper case letter "P" is 80.
- The code for letter 'a' in ASCII is 97.
- The ASCII code for the letter 'k' is 107.
- It is interesting to know that the ASCII code for the letter 'i' is 105.
- In the ASCII code system, the letter 's' has a code of 115.
- The code for 't' is 116 in ASCII.
- The ASCII code for "a" is 97.
- The ASCII code for 'n' is 110.



Class ACTiViTy

1. Write down your name.
2. Find the ASCII code for each letter in your name. You can use the ASCII table for your help.
3. Convert each ASCII code to binary.
4. Write down your name in binary!

Character	ASCII Code	Character	ASCII Code	Character	ASCII Code
SP (space)	32	-	45	:	58
!	33	.	46	;	59
"	34	/	47	<	60
#	35	0	48	=	61
\$	36	1	49	>	62
%	37	2	50	?	63
&	38	3	51	@	64
'	39	4	52	A	65
(	40	5	53	B	66
)	41	6	54	C	67
*	42	7	55	D	68
+	43	8	56	E	69
,	44	9	57	F	70



Character	ASCII Code	Character	ASCII Code	Character	ASCII Code
G	71	Y	89	k	107
H	72	Z	90	l	108
I	73	[	91	m	109
J	74	\	92	n	110
K	75	]	93	o	111
L	76	^	94	p	112
M	77	-	95	q	113
N	78	?	96	r	114
O	79	a	97	s	115
P	80	b	98	t	116
Q	81	c	99	u	117
R	82	d	100	v	118
S	83	e	101	w	119
T	84	f	102	x	120
U	85	g	103	y	121
V	86	h	104	z	122
W	87	i	105	{	123
X	88	j	106		124



Character	ASCII Code	Character	ASCII Code	Character	ASCII Code
}	125	~	126	DEL	127

Table: 1.3

Extended ASCII

While the standard ASCII Table includes 128 characters, there is an extended version that includes 256 characters. This extended ASCII uses 8 bits and includes additional symbols, accented letters, and other characters. However, the original 128 characters are the most commonly used and serves as the basis for text representation in computers.

Unicode

Unicode is an attempt at mapping all graphic characters used in any of the world's writing system. Unlike ASCII, which is limited to 7bits and can represent only 128 characters, Unicode can represent over a million characters through different forms of encodings such as, UTF-8, UTF-16, and UTF-32. UTF is an acronym that stands for Unicode Transformation Format.

UTF-8

It is a variable-length encoding scheme, meaning it can use a different numbers of bytes (from 1 to 4) to represent a character. UTF-8 is backward compatible with ASCII. It means it can understand and use the older ASCII encoding scheme without any problems. Therefore, if we have a text file written in ASCII, it will work perfectly fine with UTF-8, allowing it to read both old and new texts.

Example: The letter 'A' in Unicode, represented as, U+0041, is 01000001 in the binary format and occupies 8 bits or 1 byte.

Let's look at how Urdu letters are represented in UTF-8:

Example: The Urdu letter 'ب' is represented in Unicode as U+0628; its binary format is 11011000 10101000, means it takes 2 bytes.

UTF-16

UTF-16 is another variable character encoding mechanism, although it uses either 2 bytes or 4 bytes per character at most. Unlike UTF-8, it is not compatible with ASCII, meaning it cannot translate ASCII code.

Example: The letter A in UTF-16 is equal to 00000000 01000001 in binary or 65 in decimal (2 bytes).

For Urdu:

Example: The right Urdu letter 'ب' in UTF-16 is represented as is 00000110 00101000 in binary, which occupies 2 bytes of memory.



UTF-32

UTF-32 is a method of encoding that uses a fixed length, with all characters stored in 4 bytes per character. This makes it very simple but at the same time it may look a little complicated when it comes to space usage.

Example: Alphabet letter 'A' in UTF-32 is represented in binary as 00000000 00000000 01000001 which is 4 bytes.

How Computers Store Files

All these files (images, audio, and video) are stored as **binary data**, which means they are represented by sequences of 0s and 1s.

Storage Devices:

- **Hard Disk Drive (HDD):** Uses spinning disks to read/write data. They offer large storage capacities.
- **Solid State Drive (SSD):** Uses flash memory for faster access times and better performance.
- **Cloud Storage:** Stores files on remote servers accessible via the internet, providing flexibility and backup options.

Did You Know?

Data size is usually expressed in byte and its multiples.

- 1 Byte (B) = 8 Bits
- 1 Kilobyte (KB) = 1024 Bytes
- 1 Megabyte (MB) = 1024 Kilobytes
- 1 Gigabyte (GB) = 1024 Megabytes
- 1 Terabyte (TB) = 1024 Gigabytes
- 1 Petabyte (PB) = 1024 Terabytes
- 1 Exabyte (EB) = 1024 Petabytes
- 1 Zettabyte (ZB) = 1024 Exabytes
- 1 Yottabyte (YB) = 1024 Zettabytes



Summary

- A system is a collection of parts that work together to achieve a common goal.
- Components are the building blocks of any system. Each component plays a specific role and contributes to the overall functionality of the system.
- The environment of a system includes everything external to the system that interacts with it. It consists of all external factors that affect system's operation.
- Computer science is the study of how computers work. It looks at what computers can do and what limitations they have.
- A computer is a complex system designed to process data and perform tasks according to a set of instructions.
- The Von Neumann architecture involves several key steps for a CPU to execute instructions, including fetching, decoding, executing, and storing.
- System software is the basic software that helps a computer run and manage its hardware and software resources.
- Application software is the software designed to help users perform specific tasks or activities.
- In computing, numbering systems are crucial as they form the foundation for representing, storing, and processing information.
- Decimal number system is a number system in which base is 10 and the digits involved are 0 to 9, which are commonly used in our daily lives.
- Binary is a base-2 number system that comprises of only the digits 0 and 1. Each digit represents a power of two.
- The Octal number system is another number system that has eight as its base; thus, it has eight digits 0 to 7. Each digit represents a power of 8.
- The Hexadecimal numbering system is another type of number system with base of 16, where the number 0 to 9 and alphabets A-F are used.
- Arithmetic operations mean addition, subtraction multiplication, and division performed on numbers in given base.
- ASCII is an acronym for American Standard Code for Information Interchange. It is an industry standard used to encode text in computers and other devices.



EXERCISE

Multiple Choice Questions

1. The primary function of a system is:

- a) To work independently
- b) To achieve a common goal
- c) To create new systems
- d) To provide entertainment

2. One of the fundamental concepts of any system is:

- a) Its size
- b) Its objective
- c) Its age
- d) Its price

3. An example of a simple system is:

- a) A human body
- b) A computer network
- c) A thermostat regulating temperature
- d) The Internet

4. The basic components of a system are:

- a) Users, hardware, software
- b) Objectives, components, environment, communication
- c) Inputs, outputs, processes
- d) Sensors, actuators, controllers

5. Which of the following describes the Von Neumann architecture's main characteristic?

- a) Separate memory for data and instructions
- b) Parallel execution of instructions
- c) Single memory store for both program instructions and data
- d) Multiple CPUs for different tasks

6. ASCII stand for:

- a) American Standard Code for Information Interchange
- b) Advanced Standard Code for Information Interchange
- c) American Standard Communication for Information Interchange
- d) Advanced Standard Communication for Information Interchange

7. How many bits are used in the standard ASCII encoding?

- a) 7 bits
- b) 8 bit
- c) 16 bits
- d) 32 bits

8. How many bytes are used to store a typical integer?

- a) 1 byte
- b) 2 bytes
- c) 4 bytes
- d) 8 bytes

9. Which software is used to enhance system performance and security?

- a) Operating system
- b) Utility software
- c) Application software
- d) Device drivers

10. Which of the following is an example of application software?

- a) Microsoft Word
- b) BIOS
- c) Disk Cleanup
- d) Device Manager



Short Questions:

1. Define a system.
2. Enlist the main components of the Von Neumann architecture?
3. Enlist the four main steps in the Von Neumann architecture's instruction cycle?
4. What is a key advantage of the Von Neumann architecture?
5. What is the primary purpose of the ASCII encoding scheme?
6. Define Unicode.
7. How does the number of bits affect the range of integer values?
8. Define system software and provide two examples.
9. Differentiate between system software and application software.
10. Add 1100_2 and 1011_2
11. Subtract 0011_2 from 1010_2

Long Questions

1. Describe the basic concept of a system.
2. Explain the Von Neumann Architecture of a computer.
3. Explain the working of Von Neumann Architecture.
4. Explain how characters are encoded using Unicode also provide examples.
5. Describe in detail how integers are stored in computer memory.
6. Perform the following binary arithmetic operations:
 - a. Multiplication of 101_2 by 11_2
 - b. Division of 1100_2 by 10_2



UNIT 2

System Design and Troubleshooting



Student Learning Outcomes

By the end of this chapter, students will be able to:

- Understand Analog and Digital signals along with ADC and DAC.
- Understand Boolean functions and operations, such as Boolean AND, and OR.
- Construct Boolean expressions using variables and Boolean operators.
- Relate common Boolean identities and Boolean simplification procedures.
- Introduce several types of gates and their functions.
- Build truth tables for the operations of logical expressions.
- Introduce logic diagrams of digital system.
- Explain the importance of troubleshooting in maintaining and operating computer systems effectively.
- Describe how systematic troubleshooting helps prevent and resolve computer issues.
- Demonstrate the ability to systematically troubleshoot computer issues
- Apply basic troubleshooting techniques, including Identifying and addressing basic hardware and software issues.
- Implement strategies for maintaining a safe and organized computer workspace.
- Understand the importance of data backups and apply methods for effective data protection
- Diagnose and address hardware issues, including hardware failures like RAM or hard drive issues. Performing component replacements and upgrades.

Introduction

In this chapter, we will discuss the Boolean functions, logic, digital logic, and difference between analog and digital signals. We will also discuss several types of gates and their truth tables. At the completion of this chapter, you should be able to construct Boolean expressions, create truth table and understand the basics of digital logic.



System troubleshooting is a vital skill for keeping computers, machines, and other equipment running smoothly. When things go wrong, troubleshooting helps to identify the problem and find a quick solution, preventing issues like downtime, reduced productivity, and potential damage. For instance, if your computer suddenly stops working, knowing how to troubleshooting can help fix it without needing costly professional help.

2.1 Basic Concept of Digital Systems

Digital systems are the backbone of today's electronics and computing systems. They manipulate digital information in the form of binary digits, which are either 0 or 1 and are used in calculating devices such as calculators and computers, among others.

Analog Signal and Digital Signal

Analog signals are signals that changes with time smoothly and continuously. They can have any value within given range. Examples include voice signal (speaking), body's temperature and radio-wave signals. **Digital signals** are the signals which have only two values that are in the form of '0' and '1'. These are utilized in digital electronics and computing systems. Digital logic circuits use these binary values to perform various operations, and they are essential to the functioning in operation of computers and many other electronic devices.

Analog to digital converter (ADC) and digital to analog converters (DAC) are important operations in today's technological developments, enabling the transmission and control of signals.

Analog Signal	Digital Signal
Continuous Infinite possible values Example: Sound waves	Discrete Finite (0 or 1) Example: Binary data in computers

Analog to Digital Conversion (ADC): ADC is the conversion of analog signals into digital signals, which are discrete and can be easily processed by computerized devices like computers, smart phones and microphones.



Digital to Analog Conversion (DAC): DAC is the conversion whereby Digital signals are converted to Analog signals, making it possible for human to perceive the information, for instance through speakers, as depicted in figure 2. 1.

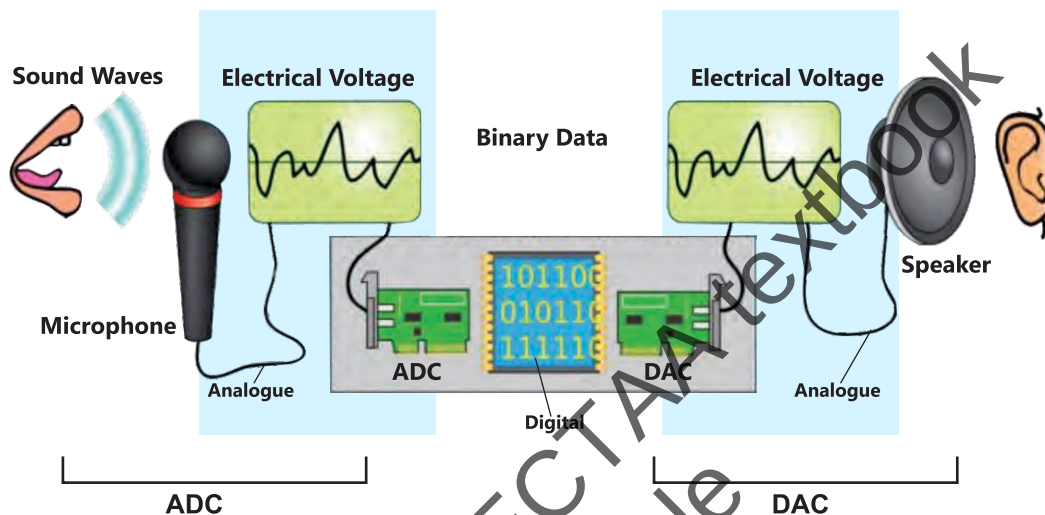


Figure 2.1: Analog to Digital and Vice Versa

ADC and DAC Conversion: Why is it needed?

Digital to analog conversion and analog to digital conversion, is critical since it enables data processing, storage, and transmission. Digital signals are much less affected by noise and signal degradation and are therefore better suited for transmitting and storing information over long distances.

Example: Sound Waves

Let us consider a situation where one person is speaking into a microphone while the other person is receiving sound through speakers as illustrated in the figure 2.1.

2.2 Boolean Algebra and Logic Gates

Boolean algebra is a branch of mathematics relate to logic and symbolic computation, using two values namely True and False. It is an essential branch of digital circuits since it is the basis for the analysis and design of circuits. Here in this section we will cover Boolean functions and expressions, the working, and functions of logic gates, Building and evaluating Truth Tables and Logic Diagrams.

Boolean Functions and Expressions

Binary values are used to describe the relationship between variables in the Boolean function and Boolean expressions.



The expressions are built using AND, OR, and other logic operations and can be reduced to optimize digital circuits in several ways.

Binary Variables and Logic Operations

Binary variables that can have only two values, 0 and 1. Logic operations are basic operations implemented in Boolean algebra for processing of these binary variables. The primary logic operations are AND, OR and NOT.

AND Operation:

AND is the basic logical operator which is used in Boolean algebra. It requires two binary inputs which will give a single binary output. The symbol '·' is used for the AND operation. The output of the AND operation is "1" only when both inputs are "1". Otherwise, the result is "0".

Example:

Consider two binary variables:

A = 1 (True)

B = 0 (False)

The AND operation for these variables can be written mathematically as:

$$P = A \cdot B$$

In this example:

A = 1, B = 0

Therefore, then, the result P of the AND operation is 0 (false).

Truth Table:

A truth table is useful in demonstrating the functionality of the AND operation with all possible inputs of the variables. Below is the truth table for the AND operation.

A	B	A AND B (P)
0	0	0
0	1	0
1	0	0
1	1	1

Table 2.1: Truth Table for AND Operation

Explanation:

If both A and B are 0, that is equal to 0 then the desired output P is (0).

if A is 0 and B is 1 the output P is 0.

When A is 1 and B is 0 P is resulting 0.

When A is 1 and B is 1, the output P also becomes 1.



OR Operation:

The OR is another basic logical operator in Boolean algebra. To be specific this is also a function that takes two binary variables as input and produces a single binary output. According to Table 2.2, the OR operation yields true (1) output when at least one of the inputs is true (1). The output is 0 only when both inputs are '0'.

Example:

Consider two binary variables:

A = 1 (true)

B = 0 (false)

The OR operation for these variables can be written mathematically as:

$$P = A + B$$

In this example:

A = 1 B = 0

Therefore, result P of the OR gate will be 1.

Truth Table:

A truth table is useful for better understanding of how the OR operation is organized and what the result of the OR's application is for all variants of the input variables. Table 2.2 shows the truth table for the OR operation.

A	B	A OR B (P)
0	0	0
0	1	1
1	0	1
1	1	1

Table 2.2: Truth Table for OR Operation

Explanation:

If A is equal to 0 and B is equal to 0 the output P is equal to 0. When A is zero and B is one, the output P is also one. When A is equal to 1 and B is equal to 0 the values of P equal to 1. When both A and B are 1 then the output P equal to 1.

Did You Know?

In binary logic, $1 + 1$ does not equal 2 but equals 1 in logical operation. This is because the OR operation returns a value of 1 if any or both of the inputs to this operator are 1.

NOT Operation:

The NOT operation is one of the basic Boolean algebra operations which takes a single binary variable and simply negates its value. If the input is one, the output is zero and if the input is zero, the output is one.



Example:

Consider a binary variable:

$$A = 1 \text{ (true)}$$

The NOT operation for this variable can be written mathematically as:

$$P = \overline{A} \quad \text{or} \quad P = \neg A$$

In this example:

$$P = 0$$

This signifies that if you have $A = 1$ (true), the result of NOT operation is going to be 0 (false).

Truth Table:

The table 2.3 illustrates the working of NOT operation for all possible inputs of the variable.

A	NOT A (P)
0	1
1	0

Table 2.3: Truth Table for NOT Operation

Explanation:

When the input A is 0, the output P is 1. When A is 1 the output value P is 0. A NOT operation performs the negative of the input variable i.e., it gives the opposite value. This operation is important in digital logic design to generate more complex logic functions and verify the functionality of digital circuits.

Construction of Boolean Functions

Boolean functions are algebraic statements that describe the relationship between binary variables and logical operations. These functions are particularly important for digital logic design and are employed in formation of various digital circuits, which are the basis of current computers, mobile phones and even simple calculator.

Understanding the Boolean Functions:

A Boolean function is a function which has a one or more binary inputs and produces a single binary output. The inputs and outputs can only have two values: False (represented by 0) and True (represented by 1). The construction of Boolean functions is done by employing the basic logical operations such as AND, OR and NOT, which connect the inputs to generate the correct output.

Example 1: Simple Boolean Function

Consider a Boolean function with two inputs, A and B. We can construct a function F that represents the AND operation:



$$F(A, B) = A \cdot B$$

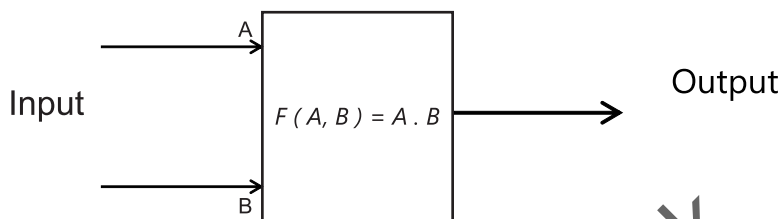


Figure 2.2: Simple Boolean Function

The diagram shown above demonstrates a basic digital circuit, which is an AND gate. The box symbolizes the AND function $F(A, B) = A \cdot B$. This box has two inputs A and B. If both A and B are 1, the output will be 1. In any other case, the output will be 0. The input are shown at the entrance to the box, while the output is depicted at the exit of the block. The truth table for this function is as follows:

A	B	F(A, B)
0	0	0
0	1	0
1	0	0
1	1	1

Table 2.4: Truth Table for $F(A, B) = A \cdot B$

Example 2: Now, let us construct a more complex Boolean function with three inputs, A, B, and C:

$$F(A, B, C) = A \cdot B + \bar{A} \cdot C$$

This function uses AND, OR and NOT at the same time. The truth table for this function is as follows:

Explanation:

- The parameters A, B, and C are included in the following example as the input columns.
- The results of AND operation between two variable A and B are presented in the column $A \cdot B$.
- The column $\bar{A}$ standing for the NOT operation of A.
- Every value in the column $\bar{A} \cdot C$ displays the result of AND operation between the values in the Fifth column and the third column.
- The final column $F(A, B, C)$ shows the output of the Boolean function $(A \cdot B) + (\bar{A} \cdot C)$

A	B	C	$A \cdot B$	$\bar{A}$	$\bar{A} \cdot C$	F(A, B, C)
0	0	0	0	1	0	0
0	0	1	0	1	1	1
0	1	0	0	1	0	0
0	1	1	0	1	1	1
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	0	1
1	1	1	1	0	0	1

Table 2.5: Detailed Truth Table for $F(A, B, C) = (A \cdot B) + (\bar{A} \cdot C)$



Usage in Computers:

- There are many uses of Boolean functions in the computers for various operations. Here are some examples of their usage:
- **Arithmetic Operations:** Boolean functions are used in Arithmetic and Logic Units (ALUs) of CPUs to perform operations like addition, subtraction, multiplication, and even division.
- **Data Processing:** Boolean functions are used to process binary data in memory and storage devices, ensuring efficient data manipulation and retrieval.
- **Control Logic:** Boolean functions are applied in computers to control various parts of the system's operation to function in co-ordinated manner.

Class ACTIVITY



Consider what do you do with your cell phone or calculator on daily basis. Can you distinguish activities that require logical choices, like entering a password to unlock your smart phone or solving a math problem? Ask your group members how Boolean functions may be utilized in the background.

Did You Know?

George Boole, a mathematician who invented Boolean algebra was born in Lincoln, England in the year 1815. His work laid the debate and the basis for future digital revolution and computer science as well as subsequent technologies of the future.

Logic Gates and their Functions

Logic gates are physical devices in electronic circuits that perform Boolean operations. Each type of logic gate corresponds to a basic Boolean operation. Examples of the logic gates are:

AND Gate: Implements the AND function. It outputs true only when both inputs are True
(1)

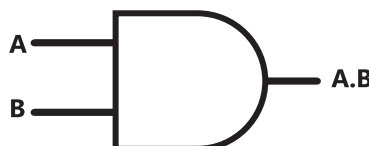


Figure 2.3: AND Gate



Imagine a simple electronic circuit with an AND gate. If you press two switches (both must be ON), a light bulb will turn on.

- Switch 1: ON (True)
- Switch 2: ON (True)
- Light bulb: ON (True) because both switches are ON.

If either switch is OFF, the light bulb will be OFF.

OR Gate: Implements the OR function. It outputs true when at least one input is true. See figure 2.4.



Figure 2.4: OR Gate

NOT Gate: Implements the NOT function. It outputs the opposite of the input. See Figure 2.5

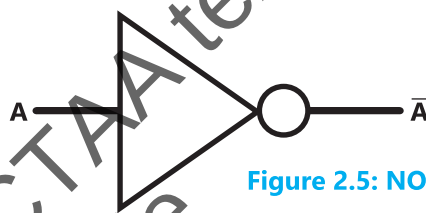


Figure 2.5: NOT Gate

NAND Gate: This gate is achieved when an AND gate is combined with a NOT gate. It generates true when at least one of the inputs is false. In other words, it is the inverse of the AND gate, as presented in Figure 2.6.

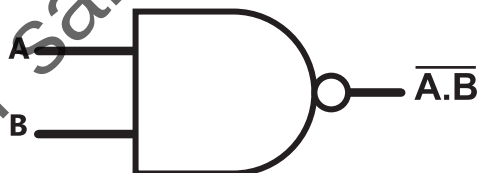


Figure 2.6: NAND Gate

Example:

Imagine a safety system where an alarm should go on if either one of two sensors detects an issue.

- Sensor 1: No issue (False)
- Sensor 2: Issue detected (True)
- Alarm: ON (True) because one sensor detects an issue.

XOR Gate:

The XOR (Exclusive OR) gate outputs true only when exactly one of the inputs is true. It differs from the OR gate, that it does not output true when both inputs are true. It is shown in Figure 2.7.

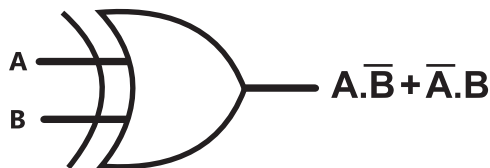


Figure 2.7: XOR Gate



Example:

Imagine a scenario where you can either play video games or do homework, but not both at the same time.

- Play video games: Yes (True)
- Do homework: No (False)
- Allowed? Yes (True) because only one activity is being done.

Class ACTiViTy



Let's make learning these logical functions fun with an activity!

1. **AND Adventure:** Form pairs and give each pair two conditions they need to meet to win a prize (like both wearing a specific color shirt).
2. **OR Options:** Make a list of fun activities. If at least one activity is possible, the class gets extra playtime.
3. **NOT Negatives:** Ask true/false questions and have students shout the opposite answer. For example, "Is the sky green?" Students should shout "No!" (NOT True).
4. Construct a basic circuit using a breadboard, a battery, and LED lights to represent an AND gate. Connect two switches which will serve as, inputs A and B. In this experiment the LED will light up only when both switches are pressed.

2.3 Simplification of Boolean Functions

Simplification of Boolean functions is a particularly important process in designing an efficient digital circuit. Such simplified functions require fewer gates making them compact in size, energy efficient and faster than the complicated ones. Simplification means applying of some Boolean algebra rules to make the functions less complicated.

Basic Boolean Algebra Rules:

Here are some fundamental Boolean algebra rules used for simplification:

1. **Identity Laws**
 $A + 0 = A$
 $A \cdot 1 = A$
2. **Null Laws**
 $A + 1 = 1$
 $A \cdot 0 = 0$



3. **Idempotent Laws** $A + A = A$
 $A \cdot A = A$
4. **Complement Laws** $A + \overline{A} = 1$
 $A \cdot \overline{A} = 0$
5. **Commutative Laws** $A + B = B + A$
 $A \cdot B = B \cdot A$
6. **Associative Laws** $(A + B) + C = A + (B + C)$
 $(A \cdot B) \cdot C = A \cdot (B \cdot C)$
7. **Distributive Laws** $A \cdot (B + C) = (A \cdot B) + (A \cdot C)$
 $A + (B \cdot C) = (A + B) \cdot (A + C)$
8. **Absorption Laws** $A + (A \cdot B) = A$
 $A \cdot (A + B) = A$
9. **De Morgan's Theorems** $\overline{A + B} = \overline{A} \cdot \overline{B}$
 $\overline{A \cdot B} = \overline{A} + \overline{B}$
10. **Double Negation Law** $\overline{\overline{A}} = A$

Simplification Examples

Example 1

Simplify the expression $A + \overline{A} \cdot B$.

Solution:

$$\begin{aligned} A + \overline{A} \cdot B &= (A + \overline{A}) \cdot (A + B) \\ &= 1 \cdot (A + B) \\ &= A + B \end{aligned}$$

(Distributive Law)

(Complement Law)

(Identity Law)

Example 2

Simplify the expression $A \cdot B + \overline{A} \cdot B$.

Solution:

$$\begin{aligned} A \cdot B + \overline{A} \cdot B &= \overline{A + B} + \overline{A} \cdot B \\ \overline{A + B} + \overline{A} \cdot B &= (\overline{A} + \overline{B}) \\ &= \overline{A} + \overline{B} \end{aligned}$$

(De Morgan's Theorem)

Since $\overline{A}$ is already present in $(\overline{A} \cdot \overline{B})$, we can use absorption law i.e $\overline{A} + (\overline{A} \cdot \overline{B}) = \overline{A}$



Example 3

Simplify the expression $(A + B) \cdot (A + \overline{B})$

Solution:

$$\begin{aligned}
 (A + B) \cdot (A + \overline{B}) &= A + (B \cdot \overline{B}) \\
 &= A + 0 \\
 &= A
 \end{aligned}$$

(Distributive Law)

(Complement Law)

(Identity Law)

Example 4

Simplify the expression $\overline{A} + B \cdot (A + \overline{B})$

Solution:

$$\begin{aligned}
 \overline{A} + B \cdot (A + \overline{B}) &= (\overline{A} \cdot \overline{B}) + (A + \overline{B}) \\
 &= \overline{A} \cdot \overline{B} + A + \overline{B} \\
 &= \overline{A} \cdot \overline{B} + A + \overline{B} \\
 &= \overline{A} \cdot \overline{B} + A + \overline{B} \\
 &= \overline{A} \cdot \overline{B}
 \end{aligned}$$

(De Morgan's Theorem)

(Distributive Law)

(Idempotent Law)

(Idempotent Law)

2.4. Creating Logic Diagrams

The logic diagrams depict the working of a digital circuit through symbols that represent to its individual logic gates. To create a logic diagram:

- Find out the logic gates needed for the Boolean function.
- Arrange the gates to perform the operations as defined by the function of the circuit.
- Connect the inputs and the output of the gates correctly.

To summarize, knowledge of Boolean algebra and logic gates is crucial when it comes to the creation and study of digital circuits. If students understand those concepts, they can build efficient and effective digital systems.

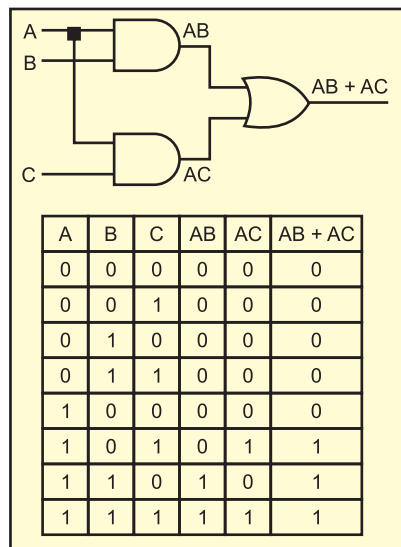


Figure 2.8: Logic Diagram



2.5 System Troubleshooting

Troubleshooting is essential for maintaining the smooth operation of systems, whether they are computers, machines, or other types of equipment. When something goes wrong, troubleshooting helps identify the problem and find a solution quickly. For example, if your computer suddenly stops working, knowing how to troubleshoot can help you get it running again without needing to call for expensive professional help.

Systematic Process of Troubleshooting

The troubleshooting process involves several steps that help you systematically identify and fix problems. These steps ensure that you don't overlook any potential issues and that you solve the problem efficiently.

A systematic approach to troubleshooting involves the following steps:

1. Identify Problem
2. Establish a Theory of Probable Cause
3. Test the Theory to Determine the Cause
4. Establish a Plan of Action to Resolve the Problem
5. Implement the Solution
6. Verify Full System Functionality
7. Document Findings, Actions, and Outcomes

1. Identify Problem

The first step in troubleshooting is to identify the problem. This means recognizing that something is not working as it should. For example, if you press power button and your laptop does not turn on, the problem is clear that it won't start.

2. Establish a Theory of Probable Cause

Once you have identified the problem, the next step is to come up with a theory about what might be causing it. This involves thinking about what could have gone wrong. For example, if your laptop does not turn on, possible causes might be a dead battery, a faulty power cord, or an internal hardware issue.

3. Test the Theory to Determine the Cause

After establishing a theory, you need to test it to see if it is correct. This involves checking if the suspected cause is actually the reason for the problem. For example, if you think the laptop's battery is dead, you can test this theory by plugging in the power cord and seeing if the laptop turns on.



4. Establish a Plan of Action to Resolve the Problem

If your test confirms the cause of the problem, the next step is to come up with a plan to fix it. This means deciding what steps you need to take to resolve the issue. For example, if the problem is a dead battery, your plan of action might be to replace the battery or keep the laptop plugged in until you can get a new one.

5. Implement the Solution

Once you have a plan, you need to put it into action. This means doing whatever is necessary to fix the problem. For example, if your plan is to replace the battery, you would buy a new battery and install it in your laptop.

6. Verify Full System Functionality

After implementing the solution, you need to check to make sure that the problem is fully resolved and the system is working properly again. For example, once you replace the battery, you should check to see if the laptop turns on and operates as expected without plugging in the power cord.

7. Document Findings, Actions, and Outcomes

The final step is to document everything you did during the troubleshooting process. This includes what the problem was, what you thought was causing it, what you did to fix it, and the outcome. This documentation is important for future reference and can help you or others troubleshoot similar problems more efficiently in the future.

By following the above steps, you can troubleshoot problems systematically and effectively, ensuring that systems continue to operate smoothly and efficiently.



Tips

Instant Fix: Sometimes, the quickest fix for a computer problem is to restart it. This is like giving the computer a nap—sometimes it just needs a short break to work properly again.

Importance of Troubleshooting in Computing Systems

Troubleshooting is very important in computing systems because it helps keep our computers, software, and networks running smoothly. When something goes wrong with a computer system, it can disrupt our work, cause data loss, or even lead to security issues. By knowing how to troubleshoot computing



systems, we can quickly find and fix problems, ensuring that everything works as it should.

- **Preventing Downtime**

Downtime occurs when a computer system is not operational. This can be very costly, especially in businesses that rely on their systems to operate efficiently. When a system is down, employees may not be able to work, leading to lost productivity and revenue. With the help of troubleshooting, we identify and resolve the system problems in very short time which facilitate to reduce the possibilities of system downtime.

- **Ensuring Data Integrity**

Data integrity means ensuring that data is accurate and reliable. Problems like software bugs or hardware failures can corrupt data, leading to incorrect information being stored or processed. Troubleshooting helps identify the source of data corruption and prevent it from happening again, ensuring that data remains accurate and reliable.

- **Improving Security**

Computer systems are often targets for cyber-attacks. Troubleshooting can help identify weakness and security breaches, allowing for quick action to protect the system. This is important for maintaining the confidentiality, integrity, and availability of data.

- **Enhancing Performance**

Sometimes, computer systems do not perform as efficiently as they should. Troubleshooting can identify the reasons for slow performance, such as insufficient memory, software conflicts, or hardware malfunctions. By resolving these issues, you can improve the overall performance of the system.

- **Extending Equipment Life**

Regular troubleshooting and maintenance can help extend the life of computer equipment. By identifying and fixing small issues before they become big problems, you can prevent unnecessary wear and tear on the system.

- **Saving Costs**

Real-World Impact: Effective troubleshooting in businesses helps prevent costly downtime and maintains productivity, showing how important these skills are in the real world. Effective troubleshooting can save money by reducing the need for expensive repairs or replacements. By identifying and resolving issues early,



you can avoid costly downtime and prolong the life of your system.

Basic Hardware-Related Issues

Common Hardware Issues and Solutions

Issue: Cable Disconnection - Loose or disconnected cables are a common hardware issue that can cause devices to stop working.

Solution: Use cable ties or organizers to keep cables neat and out of the way. Labeling cables can also help identify them easily.

Issue: Overheating - Overheating can cause a computer to slow down, freeze, or shut down unexpectedly.

Solution: Proper ventilation is crucial to prevent overheating and ensure the computer runs efficiently.

Issue: Peripheral Devices - Peripheral devices like keyboards, mouse LED can have various issues, from not being recognized by the computer to not working correctly.

Solution: Replace relevant peripheral device like keyboard, mouse or LCD.



TidBiTs

Cool Tools: Modern troubleshooting involves using special tools like memory diagnostic apps (e.g., MemTest86) and hard drive health checkers (e.g., CrystalDiskInfo). These tools are like having superpowers that can see inside your computer's brain problems by following clues and gathering evidence!

Hardware Diagnosis and Maintenance

Recognizing Hardware Failures

Recognizing hardware failures is necessary for maintaining a computer system's functionality. Here are some common symptoms and diagnostic techniques of RAM or Hard Drive Failures:

Issue: RAM Failures - Common signs of RAM issues include frequent system crashes, Blue Screens Of Death (BSOD), and poor performance. The computer may also fail to boot or restart randomly.



RAM Diagnostic Tools - Use built-in tools like Windows Memory Diagnostic or third-party applications like MemTest86 to check for RAM issues.

Did You Know?

Memory Failures: Faulty RAM can cause system crashes and data corruption. RAM errors can account for up to 10% of all computer crashes and Blue Screens Of Death (BSOD).

Reference: PCMag - Troubleshooting RAM Issues

Issue: Hard Drive Failures - Symptoms of hard drive failures include strange noises (like clicking), slow performance, frequent crashes, and corrupted files. The computer may also fail to boot or display error messages about the disk.

Hard Drive Diagnostic Tools - Use tools like SMART (Self-Monitoring, Analysis, and Reporting Technology) status checks, or software like CrystalDiskInfo, to monitor hard drive health.

Replacement and Upgradation of Components

Upgrading or replacing hardware components can significantly improve your computer's performance and extend its lifespan.

Replace RAM: To replace RAM, first determine the type and maximum capacity your motherboard supports. Purchase compatible RAM sticks, power off your computer, open the case, and insert the new RAM into the empty slots.

Example: If your computer is slow while multitasking (running multiple applications at a time), adding more RAM can help improve its performance.

Replacing a Hard Drive: To replace a hard drive, back up your data, purchase a compatible drive (that can work in your computer), power off your computer, open the case, disconnect the old drive, and connect the new one. After installing the new drive, you will need to reinstall the operating system and restore your data from the backup.

Example: If your hard drive is failing, replacing it with a new one can prevent data loss and restore your computer's functionality.

Security and Maintenance

Understanding and applying regular maintenance and security measures are essential for the proper and continued performance of a system. Using the following security, maintenance and troubleshooting techniques, you can



effectively maintain and enhance the performance and security of your computing systems.

Class ACTiViTy



Activity : Common Computer Issues

Objective: Recognize and resolve common computer issues.

Required Material: Computers, list of common issues, troubleshooting guides.

Activity Type: Individual

Activity Task Details:

1. **Introduction:** Briefly explain common computer issues like application freezing and unresponsive peripherals.
2. **Task:** Provide students with a list of common issues and troubleshooting guides.

Hands-on Practice: Each student will troubleshoot the issues on their computer and document the steps they took to resolve them. For example, if the mouse is unresponsive, check the USB connection or replace the batteries.

Class ACTiViTy



Activity : Security Practices

Objective: Learn and apply basic security practices.

Required Material: Computers, internet access, security software (e.g., antivirus).

Activity Type: Individual

Activity Task Details:

1. **Introduction:** Discuss common security threats and the importance of strong passwords and software updates.
2. **Task:** Each student will create a strong password for their computer, run a security scan using antivirus software, and update their operating system. For example, creating a password with a mix of letters, numbers, and symbols, and using antivirus software to scan for malware.

Documentation: Students will document the steps they took and the outcomes.



Maintaining Software

Keeping software up to date and resolving a conflict is essential for security and performance.

Installing Updates and Software Patches: Regularly installing updates and patches ensures that your software is protected against vulnerabilities and performs optimally.

Example: Updating your operating system and applications can protect your computer from security threats and fix bugs that cause crashes.

Resolving Software Conflicts: Identify and uninstall conflicting software, reinstall or update the affected applications, and check for compatibility issues.

Example: If two applications are causing system instability, removing one or updating both to the latest versions can resolve the conflict.

Did You Know?

The Power of Updates: Some updates, like those for operating systems or antivirus software, can be essential for security. For instance, the WannaCry ransomware attack in 2017 exploited a vulnerability in older Windows systems, which had been patched in a security update.

Reference: [BBC News - WannaCry Ransomware Attack](#)

Summary

- Digital systems are the basis of the present-day electronics and computing. They process digital data in form of '0' and '1'.
- Analog signals are continuous time varying signal.
- ADC (Analog to Digital Conversion) is the process of converting the continuous signals into discrete signals that can be processed by digital devices for example computers and smart phones.
- DAC (Digital to Analog Conversion) converts the digital signal back to the analog signal.
- Digital logic is the basis of all digital systems. This is the technique we use to process digital information in the form of binary numbers.



- In the case of AND operation the output is 1 only when both input values are 1. Otherwise, the output is 0.
- In an OR gate, the result is 0 only when both the input values are 0. Otherwise, the output is 1.
- The NOT operation the simplest logical operation in Boolean algebra, which accept a single binary inputs and gives its opposite as the outputs.
- Boolean functions are mathematical expressions that represent logical operations involving binary variables.
- A crucial element of digital circuit design is the logic diagram, which represents the structure of the circuit by showing connections between logic gates.
- Troubleshooting is very important for maintaining a smooth operation of systems like computers and machines.
- Troubleshooting helps identify and resolve problems quickly to prevent downtime and system damage.

EXERCISE

Multiple- Choice Questions (MCQs)

1. Which of the following Boolean expressions represents the OR operation?

- a) $A \cdot B$ b) $A + B$ c) A d) $A \oplus B$

2. What is the dual of the Boolean expression $A \cdot 0 = 0$?

- a) $A + 1 = 1$ b) $A + 0 = A$ c) $A \cdot 1 = A$ d) $A \cdot 0 = 0$

3. Which logic gate outputs true only if both inputs are true?

- a) OR gate b) AND gate c) XOR gate d) NOT gate

4. What is the decimal equivalent of the binary number 1101?

- a) 11 b) 12 c) 13 d) 14

5. What is the first step in the systematic process of troubleshooting?

- a) Establish a Theory of Probable Cause
b) Implement the Solution



- c) Identify Problem
- d) Document Findings, Actions, and Outcomes

6. Why is effective troubleshooting important for maintaining systems?

- a) It helps save money on repairs
- b) It prevents the need for professional help
- c) It ensures systems operate smoothly and efficiently
- d) It allows for more frequent system updates

7. Which step involves coming up with a theory about what might be causing a problem?

- a) Test the Theory to Determine the Cause
- b) Establish a Theory of Probable Cause
- c) Implement the Solution
- d) Verify Full System Functionality

8. After implementing a solution, what is the next step in the troubleshooting process?

- a) Document Findings, Actions, and Outcomes
- b) Test the Theory to Determine the Cause
- c) Verify Full System Functionality
- d) Establish a Plan of Action to Resolve the Problem

9. Why is troubleshooting important in computing systems?

- a) It ensures hardware components are always up to date
- b) It prevents the need for data backups
- c) It helps keep systems running smoothly and securely
- d) It eliminates the need for software updates

Short Questions

1. Define a Boolean function.
2. What is the significance of the truth table in digital logic?
3. Describe the function of a NOT gate with its truth table.
4. Write the purpose of ADC and DAC.
5. Make the truth of two variable A,B for AND operation.
6. What is the first step in the systematic process of troubleshooting?
7. After identifying a problem, what is the next step in troubleshooting?
8. Describe the importance of testing a theory during the troubleshooting process.



9. Explain what the "Implement the Solution" step entails in a troubleshooting.
10. Why is it necessary to verify full system functionality after implementing a solution?

Long Questions

1. Explain the usage of Boolean operations in computers.
2. Simplify the following Boolean function using Boolean algebra rules:
$$F(A, B) = A \cdot B + A \cdot \overline{B}$$
3. Discuss the importance of troubleshooting in maintaining the smooth operation of systems, especially computing systems.
4. Explain the systematic process of troubleshooting. Describe each step in detail.
5. Discuss the importance of documenting findings, actions, and outcomes during the troubleshooting process.
6. Describe the different data backup methods.

Web version of PECT AP textbook
Not for sale



UNIT

3

Computer Networks



Student Learning Outcomes

By the end of this chapter, you will be able to:

- Understand and explain computer networks as systems, their objectives, components, and data communication among these components.
- Understand fundamental concepts in data communication, including sender, receiver, protocol, message, and communication medium.
- Understand key concepts related to computer networks, including networking devices, network topologies, and transmission modes.
- Understand the 7-layer OSI networking model and its related protocols.
- Understand the benefits of using computer networks, such as resource sharing and data communication.
- Understand how protocols, data, packets, and network services like DNS and DHCP function in a networked environment.
- Understand different methods of network security, their advantages, and disadvantages.
- Understand real-world applications of computer networks, including various network-based services and how they are used.
- Know standard protocols involved in TCP/IP communications.
- Know key networking terms like the 7-layer OSI networking model, packet switching, circuit switching, router, TCP/IP, IP, UDP, DNS, DHCP, host, browsers, layering, encapsulation, and various protocols involved in TCP/IP communications.
- Differentiate between components of data communication.
- Differentiate networking devices and network topologies.
- Differentiate transmission modes.
- Identify and describe different types of networks using the 7-layer OSI networking model.
- Explain how data is transmitted across networks and describe the standard protocols involved.
- Define and explain the uses of protocols, data, packets, and network services like DNS and DHCP.



Introduction

In today's interconnected world, computer networks play a vital role in the functioning of societies and businesses. This chapter aims to provide a comprehensive understanding of computer networks as systems, including their components, objectives, and real-world applications.

3.1 Network as a System

A computer network is a system of linked devices and computers that may exchange data and operate together. Networks can range from small, Local Area Network (LANs, Local Area Networks) to large area network, WANs, including the Internet. Networks are arranged of various elements that work together to facilitate communication as depicted in Figure 3.1.

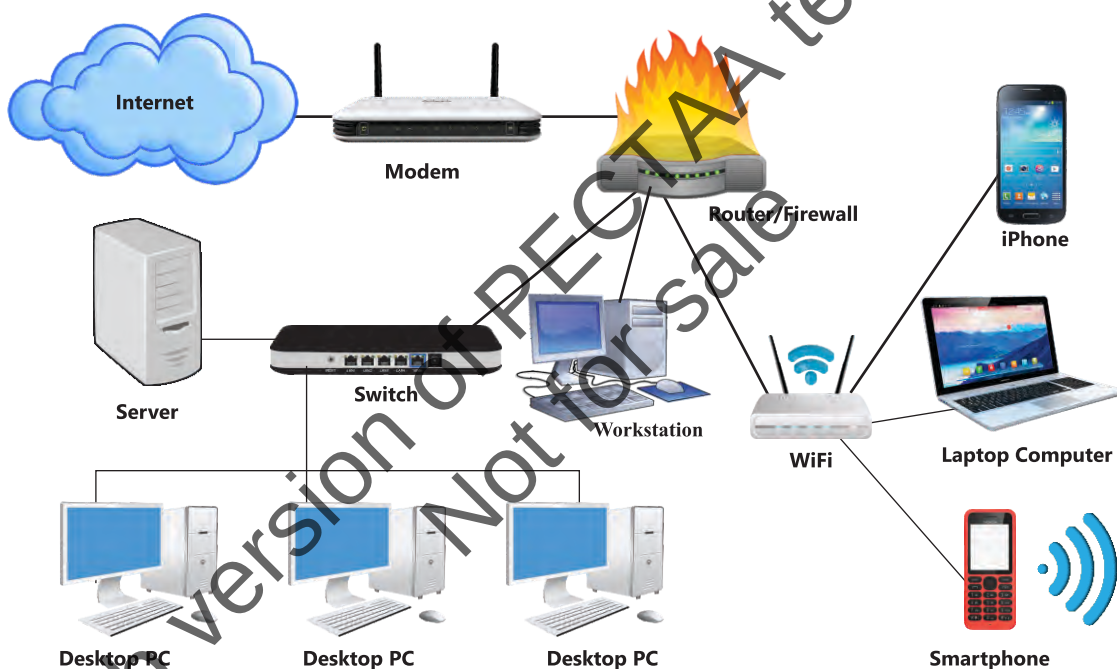


Figure 3.1: Network Diagram

The primary components include:

- **Nodes:** Devices that are connected to the network, such as computers, smartphones, and printers.
- **Links:** The connections between nodes, which can be wired (like Ethernet cables) or wireless (like Wi-Fi).
- **Switches:** Devices that connect multiple nodes within a network to forward data.



- **Routers:** Devices that connect different networks and direct data packets between them.

Did You Know?

The Internet is the largest network, connecting all the networks worldwide! It is also called network of networks.

Objectives of Computer Networks

The primary objective of computer network is to enable resource sharing, data communication and collaboration:

1. **Resource Sharing:** Computer networks allow devices to share resources, such as printers and storage, reducing costs and improving efficiency.
Example: In an office network, multiple computers can share a single printer, reducing the need for multiple printers.
2. **Data Communication:** Networks facilitate data transfer, enabling communication through emails, instant messaging, and video conferencing.
Example: Employees in different locations can collaborate through video conferencing tools like Zoom or Microsoft Teams.
3. **Connectivity and Collaboration:** Networks connect devices, allowing for remote access and collaboration, improving productivity and flexibility.
Example: A team can work on a shared document in real-time using cloud-based services like Google Drive.

Did You Know?

The World Wide Web (WWW) was invented by Tim Berners-Lee in 1989, revolutionizing how we access and share information.

3.2 Fundamental Concepts in Data Communication

Data communication involves the exchange of data between a sender and a receiver through a communication medium. Key components include the sender, receiver, message, protocol, and medium.

Components of Data Communication

It comprises of five basic components:

1. **Sender:** The device that sends the data. **Example:** A computer sending an email.
2. **Receiver:** The device that receives the data. **Example:** A smartphone receiving the email.



3. **Message:** The data being communicated. **Example:** The content of the email.
4. **Protocol:** A set of rules governing data communication. **Example:** The HTTP protocol used for web communications.
5. **Medium:** The physical or wireless path through which data travels. **Example:** Ethernet cable or Wi-Fi.

3.3 Networking Devices

Networking devices include hubs, switches, routers, and access points are responsible for the management and direction of network traffic.

Switch

Switch is a network device that connects multiple network devices such as computers, printers, and servers, within a network and allows these devices to communicate with each other efficiently. Switches play an important role in modern networks by efficiently managing data traffic and ensuring that information reaches the correct device.

How Does a Switch Work?

A switch is used at the Data Link layer which is called the Layer 2 of the OSI model (Section 3.6). It uses hardware address of a device called Media Access Control (MAC) addresses to forward data to the correct device. When a data packet reaches at the switch, it reads the destination MAC address and sends the packet only to the device with that address, rather than broadcasting it to all devices.

Example of Using Switches

- Imagine a file transfer in an office network. You send a file from your computer to a colleague's computer in another room.
- The file is split into packets, and each packet has the destination MAC address (your colleague's computer).
- The packets are sent to a network switch.
- The switch examines the Media Access Control address and forwards the packets only to the port where your colleague's computer is connected.
- Once all packets are received, your colleague's computer reassembles them into the original file.



TidBiTs

A **MAC Address** is a unique, permanent ID number assigned to every device that can connect to a network. It is used for identification directly on your local network (like in your home or office).

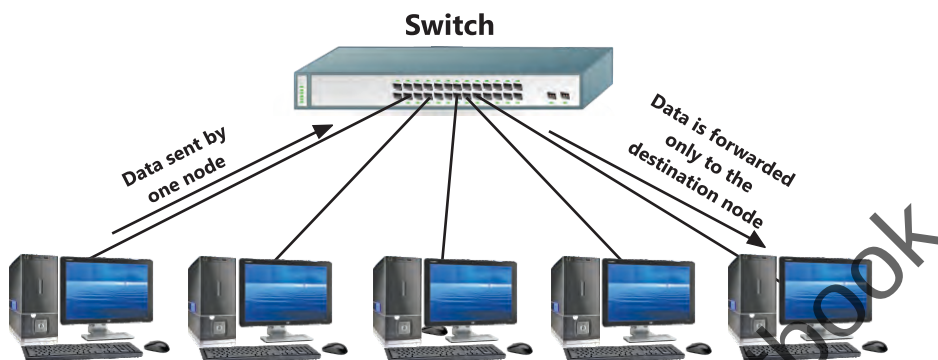


Figure 3.2: A network switch connecting multiple devices.



TidBiTs

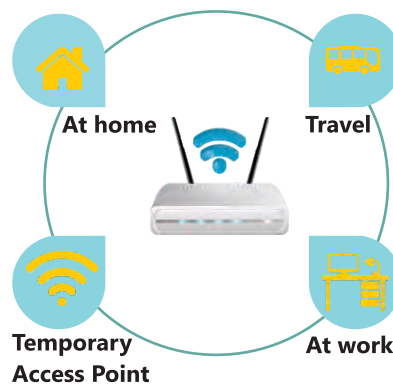
First time, switch forwards or broadcast data to all connected devices. Once it has learned address of devices, it starts sending data to exact destination.

Did You Know?

A switch is like a smart traffic conductor that directs data traffic only to the intended destination, making the network faster and more efficient.

Router

A router is a networking device that interconnects networks or allows devices to connect to it. It directs data packets between different networks. Think of it as a traffic director on the internet, making sure that data gets from one place to another efficiently. Figure 3.3 illustrates how a mobile internet connection (via SIM card) integrates with a home network. Alternatively, an Ethernet cable can be used to obtain internet access and distribute it among home devices. In enterprise environments, different types of routers are employed, as depicted in Figure 3.3.



Mobile Wi-Fi Router

Figure 3.3: A typical home router

How Does a Router Work?

When you send data on Internet it is divided into small segments called **Packets**. Each packet contains part of the data and the address of the destination. The main job of router is to find the best path for each data packet to deliver its destination.



Example: Air Travel System

Think of sending people via air travel. Here's how it relates:

When traveling, passengers (data) might be split into groups (packets) and assigned different flights (paths). In packet switching each group has a ticket with the final destination (IP address).

These groups might take different routes, through various airports (routers), to reach the final destination.

Did You Know?

Routers use something called a routing table to decide the best path for data packets. This table lists the possible paths and helps the router make efficient decisions!

Class ACTiViTy



Human Network Activity: Create a simple network using the students in the class. Assign roles like computer, router, and data packet. Use strings to represent Ethernet cables and have students pass a ball (representing data) along the strings to simulate how a router directs data.



TidBiTs

To keep your network running smoothly, always use high-quality Ethernet cables and ensure your switch is placed in a cool, ventilated area to prevent overheating.

Access Point

An **Access Point (AP)** is a networking device that facilitates the connection of wireless devices to a wired network. It works as a link between your computers and smartphones or any other wireless device and the internet.

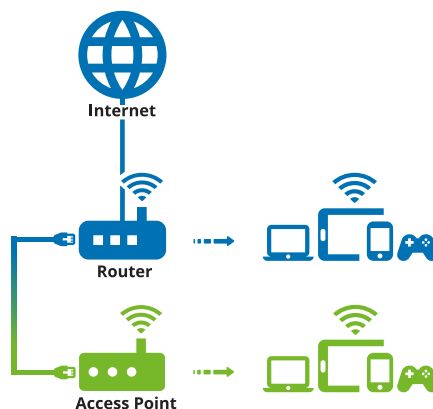


Figure 3.4: A typical Access Point



How Does an Access Point Work?

An Access Point works by receiving data from the wired network and transmitting it wirelessly to your devices. It also receives data from your wireless devices and sends it to the wired network.



TidBiTs

When setting up an Access Point, place it in a central location to ensure the best coverage and signal strength for all your devices!

3.4 Network Topologies

Network topologies are methods used to define the arrangement of different devices in a computer network, where each device is called a node. The reliability and performance of a network are impacted by the way its devices are linked.

Bus Topology

In a Bus topology, all devices share a single communication line called a bus. Each device is connected to this central cable.

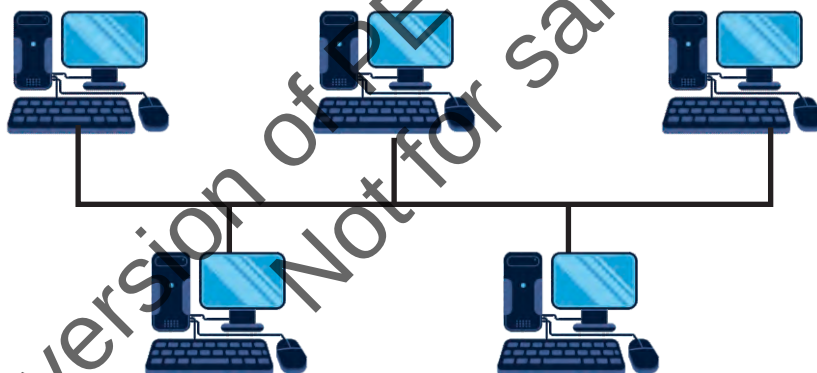


Figure 3.5: Bus Topology

Did you Know?

Bus topology is easy to set up but if the main cable fails, the whole network goes down!

Star Topology

In a star topology each node in network communicates with the others via a central switch or hub. The hub works as a data flow repeater.

Example: Think of a school principal's office connected to all classrooms through intercoms. The principal's office is the hub, and the classrooms are the nodes.

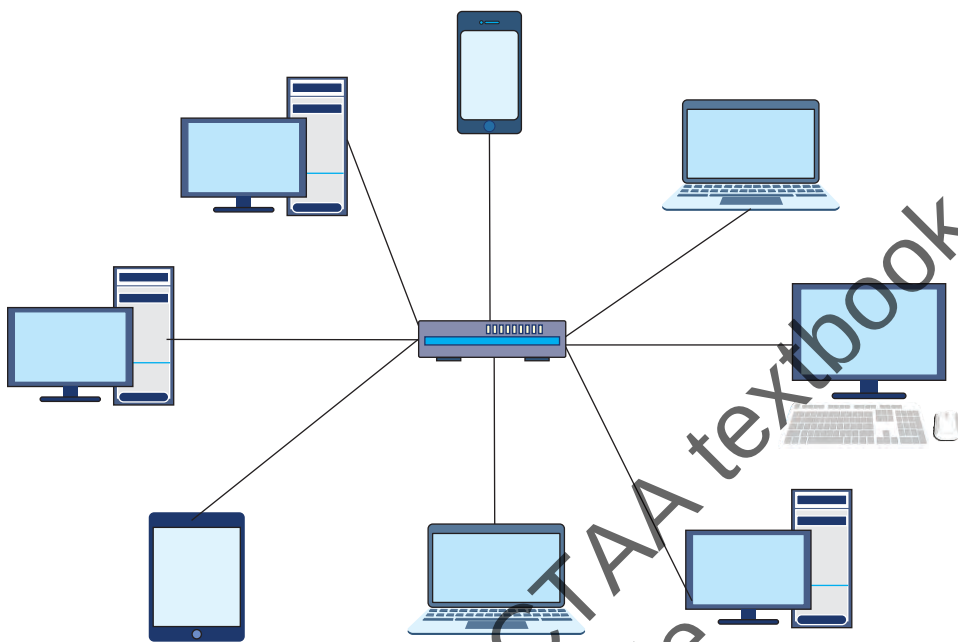


Figure 3.6: Star Topology

Class ACTiVITY



Create a star topology model using strings and paper cups to simulate the connections between a central switch and peripheral nodes.

Did You Know?

In star topology if hub/ switch stop working the whole network goes down.

Ring Topology

In a Ring topology, each device is connected in a circular pathway with two other devices. Data travels in one direction, passing through each device.

Example: Consider a relay race where each runner passes the baton to the next runner in a circle until it reaches the starting point again.

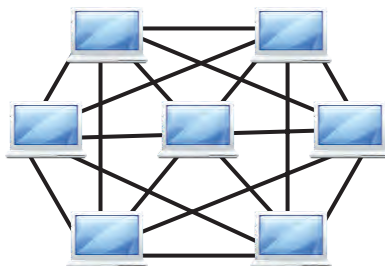


Figure 3.7 Ring Topology



TidBiTs

Ring topology can handle high traffic, but if one connection fails, the whole network is affected. Then 2-way ring can solve this issue to some extent.



Mesh Topology

In a Mesh topology, each device is connected to every other device. This provides high redundancy and reliability.

Figure 3.8: Mesh Topology

Example: Imagine a city where every house is directly connected to every other house by roads. If one road is blocked, there are multiple alternative routes.

Did You Know?

Mesh topology is very reliable because if one link fails, data can be rerouted through other links.

3.5 Transmission Modes

Network communication modes describe how data is transmitted between devices. There are three primary modes: Simplex, Half-Duplex, and Full-Duplex as shown in Figure 6.13. Let's explore each mode with examples and real-life analogies!

Simplex Communication

In Simplex communication, data transmission is unidirectional, meaning it flows in only one direction. A advice can either send or receive data in this communication.



Figure 3.9: Simplex Communication



Example: Keyboard to computer is an example of simplex communication.

Did You Know?

In Simplex communication, the direction of data flow is fixed, making it useful for applications where only one-way communication is needed!

Half-Duplex Communication

In Half-Duplex communication, data transmission can occur in both directions, but not simultaneously. One device must wait for the other to finish transmitting before it can start.

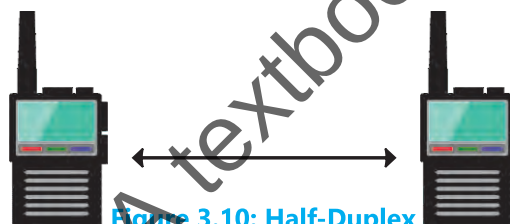


Figure 3.10: Half-Duplex Communication

Class ACTiVITy



Use walkie-talkies or toy telephones to demonstrate Half-Duplex communication. Let students take turns speaking and listening.

Full-Duplex Communication

Full-duplex communication allows for simultaneous data delivery in both directions. Both devices may transmit and receive data simultaneously at the same time.



Figure 3.11: Full-Duplex Communication

Example:

Telephone conversation is an example of Full-Duplex communication. Both people can talk and listen at the same time without waiting for their turn.

Did You Know?

The first telephones were Half-Duplex, where only one person could speak at a time. Modern phones use Full-Duplex, allowing both people to talk and listen simultaneously!



3.6 The OSI Networking Model

The Open Systems Interconnection (OSI) Model is a framework used to understand how different networking protocols interact. It has 7 layers, each with a specific function. Let's explore these layers with examples and relate them to daily life.

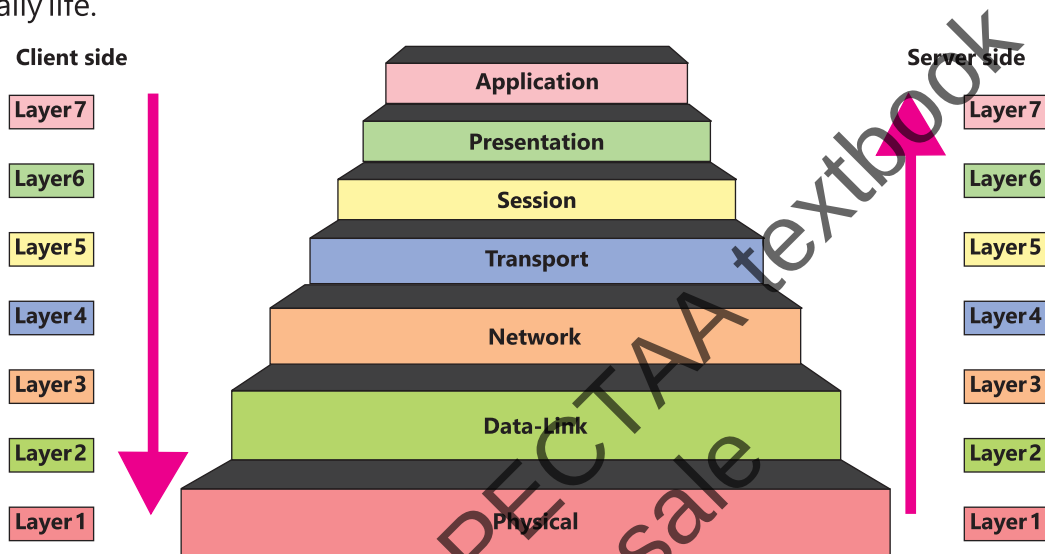


Figure 3.12: OSI Model

Layer 1: Physical Layer

The Physical Layer is liable for the actual connection between devices. The process of sending unprocessed data bits via a physical medium is the focus here.

Example: Imagine the hardware that connects computers, like a Network interface cables, repeaters, hubs and connectors.

Did You Know?

The Physical Layer includes everything from the cables to the voltage levels used to transmit data!

Layer 2: Data Link Layer

Error detection and correction, as well as node-to-node data transport, are handled by the Data Link Layer. It ensures error-free data transmission from the Physical Layer.

Example: Think of the Data Link Layer as traffic lights at intersections, which manage the flow of cars (data) and prevent collisions.



Class ACTiViTy

Draw a simple network with devices and label the physical connections and data link layer responsibilities.

Layer 3: Network Layer

The Network Layer is responsible for data transfer between different networks. It determines the best path for data to travel from the source to the destination.

Example: Imagine a GPS system finding the best route for you to travel from home to school.



TidBiTs

The Network Layer uses IP addresses to route data between networks!

Layer 4: Transport Layer

The Transport Layer ensures that data is transferred from source system to destination system. It manages data flow control and error checking.

Example: Think of the Transport Layer as a delivery service that ensures your package arrives safely and on time.

Did You Know?

The Transport Layer uses protocols like Transmission Control Protocol (TCP) to ensure reliable data transfer!

Layer 5: Session Layer

The Session Layer manages sessions between applications. It establishes, maintains, and terminates connections between devices.

Example: Imagine a phone call where the session layer sets up the call, keeps it connected, and ends it when you hang up.

Class ACTiViTy

Role-play a phone call and discuss how the session is established, maintained, and terminated.

Layer 6: Presentation Layer

The Presentation Layer translates data between the application layer and the network. It formats and encrypts data to ensure it is readable by the receiving system.



Example: Think of the Presentation Layer as a translator converting a book from one language to another so that more people can read it.

Did You Know?

The Presentation Layer handles data encryption and compression!

Layer 7: Application Layer

The Application Layer is the closest to the end user. It provides network services directly to applications, such as email, web browsing, and file transfer.

Example: Imagine the Application Layer as a waiter taking your order in a restaurant and bringing your food.

Class ACTiViTy



List the applications you use daily and identify which rely on the Application Layer for network services.

3.7 Ipv4 and Ipv6

Internet Protocol (IP) addresses are unique identifiers assigned to devices connected to the Internet. There are two primary versions: IPv4 and IPv6. Let's explore the differences between them with examples and relate them to daily life.

Internet Protocol version 4 (IPv4)

IPv4 is the fourth version of the Internet Protocol and the most widely used today. It uses a 32-bit address scheme, allowing for approximately 4.3 billion unique addresses. To find the total number of unique IPv4 addresses, we calculate 2^{32} , which represents all possible combinations of 32 bits, i.e., $2^{32} = 4,294,967,296$.

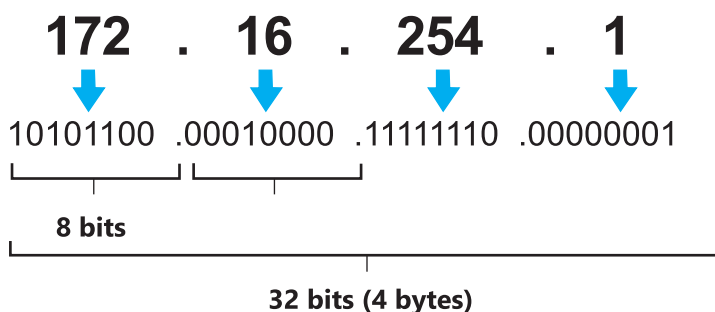


Figure 3.13: IPv4 Address Format



Did You Know?

IPv4 addresses are written in four sets of decimal numbers, each ranging from 0 to 255 (e.g., 192.168.1.1).

Internet Protocol version 6 (IPv6)

IPv6 is the most recent version of the Internet Protocol designed to replace IPv4. It uses a 128-bit address scheme, allowing for an almost limitless number of unique addresses.

Example: Imagine an IPv6 address like a digital fingerprint. It can provide a unique identifier not just for houses on a street, but for every grain of sand on a beach. e.g 2001 : 0000 : 130 F : 0000 : 0000 : 0900 : 876A : 130B

Did You Know?

IPv6 was developed to address the depletion of IPv4 addresses due to the rapid growth of the internet and connected devices.

3.8 Network Protocols

Introduction to Protocols

Protocols are sets of rules that govern data communication. Common protocols include TCP/IP, HTTP, FTP and SMTP.

Example: HyperText Transfer Protocol (HTTP) is used for transferring web pages over the internet.

DNS and DHCP

Domain Name System (DNS)

DNS translates domain names to IP addresses, making it easier for users to access websites.

Example: When you type **www.example.com** in a browser, DNS translates it to the corresponding IP address.

Dynamic Host Configuration Protocol (DHCP)

DHCP automatically assigns IP addresses to devices on a network, simplifying network management.

Example: When a device connects to a Wi-Fi network, DHCP assigns it an IP address.



Transmission Control Protocol / Internet Protocol ((TCP/IP)

TCP/IP (Transmission Control Protocol/Internet Protocol) is the fundamental suite of protocols for internet communication.

- **Transmission Control Protocol (TCP):** Ensures reliable data transfer.
- **Internet Protocol (IP):** Handles addressing and routing of data packets.
- **User Datagram Protocol (UDP):** Provides faster, but less reliable, data transfer.

3.9 Network Security

Network security involves measures to protect data and prevent unauthorized access to computer networks. Let's explore the importance of network security and some key concepts with examples.

Importance of Network Security

Network security is important for several reasons:

- **Data Protection:** Ensuring that sensitive information is not accessed or altered by unauthorized users.
- **Preventing Attacks:** Defending against malicious attacks that can disrupt networks and steal data.
- **Maintaining Privacy:** Safeguarding personal and confidential information from being compromised.
- **Ensuring Availability:** Ensuring that network resources are available and accessible to authorized users.

3.10 Network Security Methods

Firewalls

Monitor and control incoming and outgoing network traffic.

Encryption

Protects data by converting it into a secure format.

Antivirus Software

Detects and removes malicious software.

Example: A combination of firewalls, encryption, and antivirus software provides robust network security.

3.11 Types of Networks

Networks are classified based on their size, range, and purpose. Let's explore some common types of networks and understand how they work.



Personal Area Network (PAN)

A PAN is a small network used for communication between personal devices, such as smartphones, tablets, and laptops, within a short range.

Example: Bluetooth connections between a smartphone and a wireless headset form a PAN.

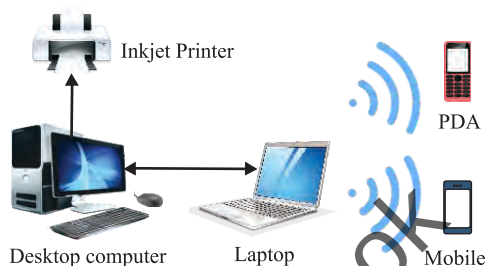


Figure No: 3.14 Personal Area Network (PAN)

Did You Know?

The range of a PAN is typically a few meters, perfect for personal device communication.

Local Area Network (LAN)

A LAN is a network that connects computers and devices within a limited area, such as a home, school, or office building.

Example: The computer network in your school that connects all the computers in the lab is a LAN.

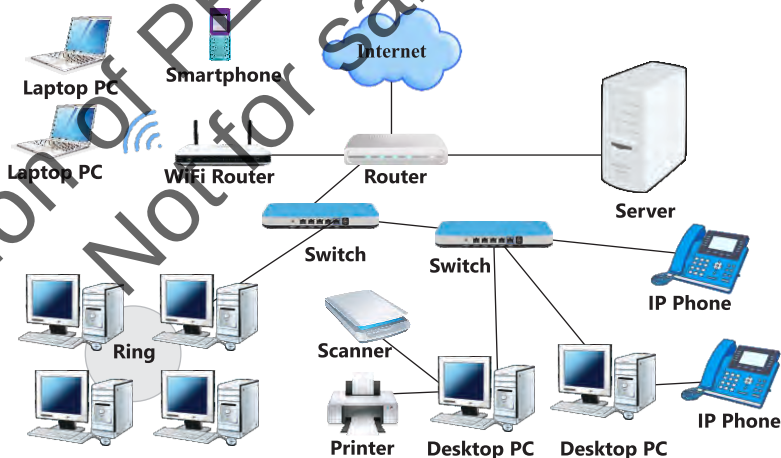


Figure 3.15: Local Area Network (LAN)

Class ACTiViTy



Draw a diagram of your school's computer network, labeling the different devices and connections.



Metropolitan Area Network (MAN)

A MAN is a network that spans a city or a large campus, connecting multiple LANs together.

Example: The network that connects various branches of a university across a city is a MAN.

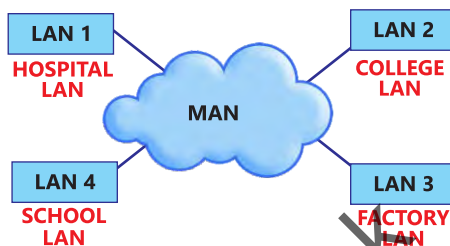


Figure 3.16: Metropolitan Area Network (MAN)

Did You Know?

A MAN can cover an area of up to 50 kilometers, making it ideal for city-wide connectivity.

Wide Area Network (WAN)

A WAN covers a large geographical area, connecting multiple LANs and MANs. The internet is the largest example of a WAN.

Example: The network that connects different branch offices of a multinational company across countries is a WAN.

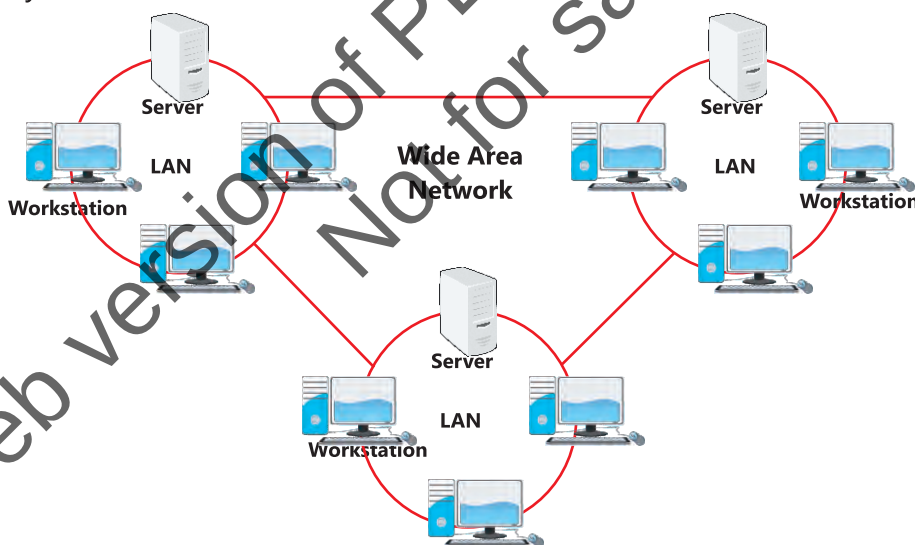


Figure 3.17: Wide Area Network (WAN)



TidBiTs

Use a Virtual Private Network (VPN) to securely connect to a WAN and protect your data when accessing public networks.



Campus Area Network (CAN)

A CAN is a network that connects multiple LANs within a limited geographical area, such as a university campus or a business park.

Example: The network that connects various departments and buildings within a university is a CAN.

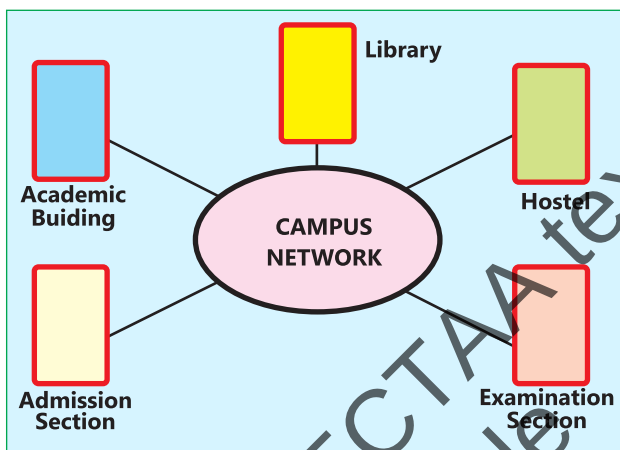


Figure 3.18: Campus Area Network (CAN)

Understanding the different types of networks helps us comprehend how data travels from one device to another, whether within a single room or across the globe. Each network type serves a specific purpose and is designed to handle various ranges and sizes.

3.12 Real-World Applications of Computer Networks

Business

In business, networks enable efficient communication, resource sharing, and data management.

Did You Know?

Intranet: A **private network** inside a company that uses internet technology (like websites). Only people **within the organization** can access it.

Example: Companies use intranets to share information and resources securely within the organization.

Education

Educational institutions use networks to provide online learning platforms, virtual classrooms, and access to educational resources.



Example: Universities use Learning Management Systems (LMS) like Blackboard and Moodle to deliver course content and assessments.

Healthcare

Healthcare networks facilitate the sharing of patient information, telemedicine, and access to medical databases.

Example: Hospitals use Electronic Health Records (EHR) systems to store and retrieve patient data efficiently.

Class ACTiVITY



Students have to discuss the advantages and disadvantages of different network security methods.

Summary

- A computer network is a system of interconnected computers and devices that communicate and share resources.
- The primary objectives of computer networks are to enable resource sharing, data communication, and connectivity between devices.
- Data communication involves the exchange of data between a sender and a receiver through a communication medium.
- A router is a device that connects different networks together and directs data packets between them.
- A switch is a network device that connects multiple devices (like computers, printers, and servers) within a Local Area Network (LAN).
- Network topologies refer to the arrangement of different elements (links, nodes, etc.) in a computer network.
- In a Bus topology, all devices share a single communication line called a bus. Each device is connected to this central cable.
- In a Star topology, all devices are connected to a central hub or switch. The hub acts as a repeater for data flow.
- In a Ring topology, each device is connected to two other devices, forming a circular data path. Data travels in one direction, passing through each device.
- In Simplex communication, data transmission is unidirectional, meaning it flows in only one direction.
- In Half-Duplex communication, data transmission can occur in both



directions, but not simultaneously.

- In Full-Duplex communication, data transmission can occur in both directions simultaneously.
- The Open Systems Interconnection (OSI) Model is a framework used to understand how different networking protocols interact.
- DNS translates domain names to IP addresses, making it easier for users to access websites.
- Network security involves measures to protect data and prevent unauthorized access to computer networks.
- A LAN is a network that connects computers and devices within a limited area, such as a home, school, or office building.
- A MAN is a network that spans a city or a large campus, connecting multiple LANs together.
- A WAN covers a large geographical area, connecting multiple LANs and MANs. The internet is the largest example of a WAN.

EXERCISE

Multiple Choice Questions (MCQs)

1. The primary objective of computer networks is:

- a) Increase computational power
- b) Enable resource sharing and data communication
- c) Enhance graphic capabilities
- d) Improve software development

2. Which device is used to connect multiple networks and direct data packets between them?

- a) Switch
- b) Hub
- c) Router
- d) Modem

3. Which layer of the OSI model is responsible for node-to-node data transfer and error detection?

- a) Physical Layer
- b) Data Link Layer
- c) Network Layer
- d) Transport Layer

4. The function of the Domain Name System (DNS) is to:

- a) Assign IP addresses dynamically
- b) Translate domain names to IP addresses
- c) Secure data communication
- d) Monitor network traffic



5. Which method of data transmission uses a dedicated communication path?

- a) Packet Switching
- b) Circuit Switching
- c) Full-Duplex
- d) Half-Duplex

6. Which protocol is used for reliable data transfer in the TCP/IP model?

- a) HTTP
- b) FTP
- c) TCP
- d) UDP

7. The main purpose of a firewall in network security is to:

- a) Convert data into a secure format
- b) Monitor and control network traffic
- c) Assign IP addresses
- d) Translate domain names

8. Which network topology connects all devices to a central hub?

- a) Ring
- b) Mesh
- c) Bus
- d) Star

9. What is a key benefit of using computer networks in businesses?

- a) Increase computational power
- b) Enable resource sharing and efficient communication
- c) Enhance graphic capabilities
- d) Improve software development

Short Questions

1. Define data communication.
2. Enlist its key components of data communications.
3. Write the role of routers in a computer network.
4. What are the main functions of the Network Layer in the OSI model?
5. What is the purpose of the Dynamic Host Configuration Protocol (DHCP)?
6. Differentiate between TCP and UDP.
7. Explain the importance of encryption in network security.
8. Define star topology?
9. How does firewall contribute to network security?
10. What is difference between Half Duplex and Full Duplex communication?

Long Questions

1. Discuss the objectives of computer networks?
2. In a Simplex communication system, assume data is transmitted at a rate of 500 bits per second (bps). Compute the time to transmit a message if:
 - a) it is of 10 kilobits.
 - b) it is of 10 kilobytes.
3. What is meant by protocols in data communication. Explain its different types.
4. Describe real-world applications of computer networks in business, education.
5. What is network topology? Explain any two with diagrams.
6. Define OSI network model, explain its any three layers.



UNIT

4

Computational Thinking



Student Learning Outcomes

By the end of this chapter, you will be able to:

- Define computational thinking and its key components: decomposition, pattern recognition, abstraction, and algorithms.
- Explain the principles of computational thinking, including problem understanding, problem simplification, and solution selection and design.
- Describe algorithm design methods, specifically flowcharts and pseudocode, and understand the differences between them.
- Create and interpret flowcharts to represent algorithms visually.
- Write pseudocode to outline algorithms in a structured, human-readable format.
- Engage in algorithmic activities, such as design and evaluation techniques.
- Conduct dry runs of flowcharts and pseudocode to manually verify their correctness.
- Understand the concept and importance of LARP (Logic of Algorithms for Resolution of Problems).
- Implement LARP activities to practice writing algorithms and drawing flowcharts.
- Identify different types of errors in algorithms, including syntax errors, logical errors, and runtime errors.
- Apply debugging techniques to find and fix errors in algorithms.
- Recognize common error messages encountered during LARP and learn how to address them.
- Demonstrate problem-solving skills by applying computational thinking principles to real-world scenarios.
- Evaluate the efficiency of different algorithms and improve them based on performance analysis.



Introduction

Computational thinking is an essential skill that enables individuals to solve complex problems using methods that align with processes involved in computer science. This chapter begins by defining computational thinking and breaking it down into its fundamental components: decomposition, pattern recognition, abstraction, and algorithms. These components are essential for simplifying complicated problems, identifying patterns that can lead to solutions, focusing on relevant details while ignoring unnecessary ones, and creating step-by-step procedures for solving problems. Understanding these concepts is not only beneficial for computer scientists but also for others looking to improve their problem-solving skills across various fields.

In addition to defining computational thinking, this chapter emphasizes the importance of practicing algorithm design and evaluation through hands-on activities like LARP (Logic of Algorithms for Resolution of Problems). By mastering these skills, students will be well-equipped to tackle a wide range of computational problems efficiently and effectively.

4.1 Computational Thinking

Computational Thinking (CT) is a problem-solving process that involves a set of skills and techniques to solve complex problems in a way that it can be executed by a computer. This approach can be used in various fields beyond computer science, such as biology, mathematics, and even in daily life.

Decomposition

Decomposition is the method of breaking down a complicated problem into smaller, more convenient components.

Decomposition is an important step in computational thinking. It involves dividing a complex problem into smaller, manageable tasks. Let's take the example of building a birdhouse. This task might look tough at first, but if we break it down, we can handle each part one at a time.

Did You Know?

Computational thinking is not limited to computer science. It is used in everyday problem solving, such as planning a trip or organizing tasks.



Here's how we can decompose the task of building a birdhouse.

- **Design the Birdhouse:** Decide on the size, shape, and design. Sketch a plan and gather all necessary measurements.
- **Gather Materials:** List all the materials needed such as wood, nails, paint, and tools like a hammer and saw.
- **Cut the Wood:** Measure and cut the wood into the required pieces according to the design.
- **Assemble the Pieces:** Follow the plan to assemble the pieces of wood together to form the structure of the birdhouse.
- **Paint and Decorate:** Paint the birdhouse and add any decorations to make it attractive for birds.
- **Install the Birdhouse:** Find a suitable location and securely install the birdhouse where birds can easily access it.

Pattern Recognition

Pattern recognition is an essential aspect of computational thinking. It involves identifying and understanding regularities or patterns within a set of data or problems. Let's consider the example of recognizing patterns in the areas of squares.

- Side Length 1: Area = $1^2 = 1$
- Side Length 2: Area = $2^2 = 4$ ($1 + 3$)
- Side Length 3: Area = $3^2 = 9$ ($1 + 3 + 5$)
- Side Length 4: Area = $4^2 = 16$ ($1 + 3 + 5 + 7$)
- Side Length 5: Area = $5^2 = 25$ ($1 + 3 + 5 + 7 + 9$)

We can see that the area of each square can be calculated by adding consecutive odd numbers. For example, the area of a square with a side length of 3 can be found by adding the first three odd numbers: $1 + 3 + 5 = 9$.

Visual/Numerical Pattern

Goes up by 1

	+1	+1	+1	+1	+1
Side	1	2	3	4	5
Area	1	4	9	16	25
	+3	+5	+7	+9	+11

Goes up by consecutive odd numbers starting at 3

Figure 4.1: Pattern in areas of squares with sides from 1 to 5



Class ACTiViTy

Create a table with side lengths from 1 to 10. Calculate the areas of the squares using the pattern of adding consecutive odd numbers. Verify your results by squaring the side lengths and see if the pattern holds.

Abstraction

Abstraction is a fundamental concept in problem solving, especially in computer science.

- **Definition:** Abstraction is the process of hiding the complex details while exposing only the necessary parts. It helps reduce complexity by allowing us to focus on the high-level overview without getting lost in the details.
- **Example:** Making a Cup of Tea - **High-level Steps:** 1. Boil water. 2. Add tea leaves or a tea bag. 3. Wait for a few minutes. 4. Pour into a cup and add milk/sugar if desired.

Algorithms

An algorithm is a step-by-step set of instructions to solve a problem or complete a task similar to following a recipe to bake a cake..

An **algorithm** is a precise sequence of instructions that can be followed to achieve a specific goal, like a recipe or a set of directions that tells you exactly what to do and in what order.

HOW TO BAKE A CAKE?

- 1) Preheat the oven
- 2) Gather the ingredients
- 3) Measure out the ingredients
- 4) Mix together the ingredients to make the batter
- 5) Grease a pan
- 6) Pour the batter into the pan
- 7) Put the pan in the oven
- 8) Set a timer
- 9) When the timer goes off, take the pan out of the oven
- 10) Enjoy!



Figure 4.2: Algorithm example: Recipe to bake a cake

- **Example :** Planting a Tree: Here is a simple algorithm to plant a tree, an activity that can be very meaningful and beneficial:
 1. Choose a suitable spot in your garden.
 2. Dig a hole that is twice the width of the tree's root ball.
 3. Place the tree in the hole, making sure it is upright.
 4. Fill the hole with soil, pressing it down gently to remove air pockets.



5. Water the tree generously to help it settle.
6. Add mulch around the base of the tree to retain moisture.
7. Water the tree regularly until it is established.

This algorithm gives clear instructions on how to plant a tree, making it easy to follow for anyone.

Did You Know?

Did you know that algorithms are not just used in computers? They are everywhere! When you follow directions to your friend's house or play a board game with rules, you are using algorithms. Algorithms help us solve problems logically.

4.2 Principles of Computational Thinking

Computational thinking involves several key principles that guide the process of problem-solving in a structured manner.

Problem Understanding

Understanding a problem involves identifying the core issue, defining the requirements, and setting the objectives. Understanding the problem is the first and most important step in problem solving, especially in computational thinking. This involves thoroughly analyzing the problem to identify its key components and requirements before attempting to find a solution.



TidBiTs

"If I had an hour to solve a problem I'd spend 55 minutes thinking about the problem and 5 minutes thinking about solutions". — **Albert Einstein**



TidBiTs

Always take time to thoroughly understand a problem before starting to solve it. Ask questions, gather information, and clarify any doubts. This core step will lead to better and more effective solutions.



Problem Simplification

Simplifying a problem involves breaking it down into smaller, more manageable sub-problems. Example: To design a website, break down the tasks into designing the layout, creating content, and coding the functionality.

Solution Selection and Design

Choosing the best solution involves evaluating different approaches and selecting the most efficient one. Designing the solution requires creating a detailed plan or algorithm.

4.3 Algorithm Design Methods

Algorithm design methods provide a range of tools and techniques to tackle various computational problems effectively. Each method has its strengths and weaknesses, making it suitable for different types of problems. Understanding different methods allows one to choose the most appropriate approach for a given problem, leading to more efficient and elegant solutions. Let's discuss two of these methods.

Flowcharts

Flowcharts are visual representations of the steps in a process or system, depicted using different symbols connected by arrows. They are widely used in various fields, including computer science, engineering, and business, to model processes, design systems, and communicate complex workflows clearly and effectively.

Importance of Flowcharts

- **Clarity:** Flowcharts provide a clear and concise way to represent processes, making them easier to understand at a glance.
- **Communication:** They are excellent tools for communicating complex processes to a wide audience, ensuring everyone has a common understanding.
- **Problem Solving:** Flowcharts help identify bottlenecks and inefficiencies in a process, aiding in problem-solving and optimization.
- **Documentation:** They serve as essential documentation for systems and processes, which is useful for training and reference purposes.



Flowchart Symbols

Flowchart symbols are visual representations used to illustrate the steps and flow of a process or system as shown in Table 4.1.

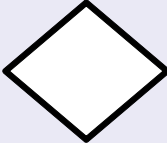
Symbol	Name	Description
	Oval (Terminal)	Represents the start or end of a process. Often labeled as "Start" or "End."
	Rectangle (Process)	Represents a process, task, or operation that needs to be performed.
	Parallelogram (Input/Output)	Represents data input or output (e.g., reading input from a user or displaying output on a screen).
	Diamond (Decision)	Represents a decision point in the process where the flow can branch based on a yes/no question or true/false condition.
	Arrow (Flowline)	Shows the direction of flow within the flowchart, connecting the symbols to indicate the sequence of steps.

Table 4.1: Flowchart symbols

Did You Know?

Flowcharts were popularized by computer scientists such as John von Neumann and Herman Goldstine in the early days of computing.



Example: A flowchart to add two numbers shows steps such as input numbers, adding the numbers and saving in "Sum" and finally displaying the "Sum". As shown in figure 4.3.

Class ACTiViTY

Draw a flowchart for selecting the school cricket team. The team can have a maximum of 11 players, and each player must have parental permission.

Example

The flowchart takes a number from user and stores it in "No." variable. At next step it takes the modulus of the No. with 2 and compares it with '0'. If condition is true (modulus is '0') then it displays that "No. is Even" otherwise it displays that "No. is Odd".

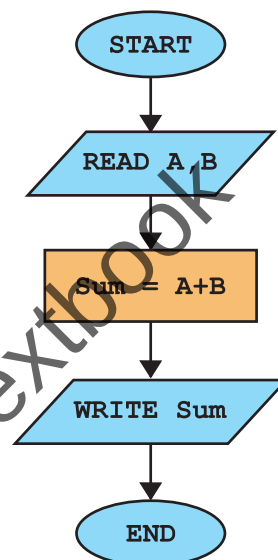


Figure 4.4 Flowchart of Adding Two Numbers

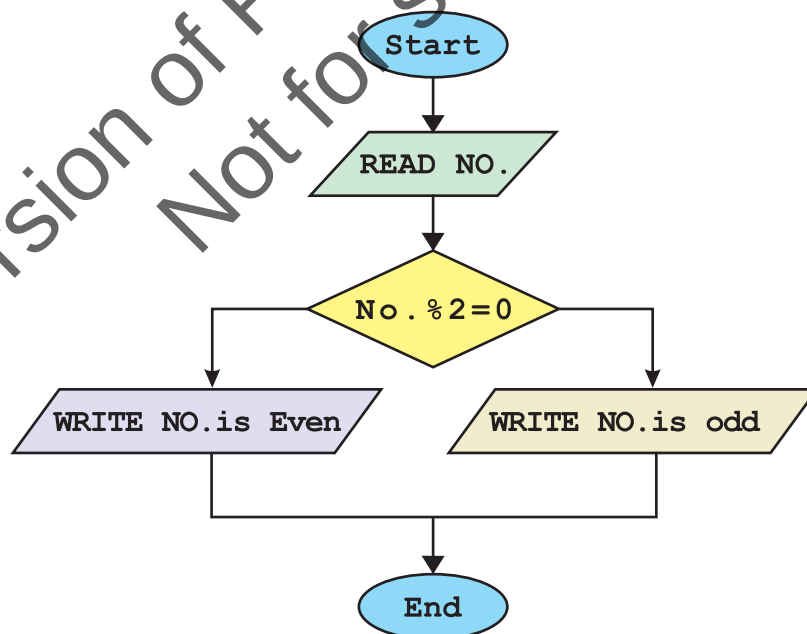


Figure 4.5: Flowchart of Finding Even or Odd Number



Example: A flowchart for a login system showing steps such as inputting a username and password, verifying credentials, and granting access shown in Figure 4.6. A user can make a maximum of five attempts.

Pseudocode

Pseudocode is a method of representing an algorithm using simple and informal language that is easy to understand. It combines the structure of programming clarity with the readability of plain English, making it a useful tool for planning and explaining algorithms.

How Pseudocode works?

Pseudocode is not actual code that can be run on a computer, but rather a way to describe the steps of an algorithm in a manner that is easy to follow. It helps programmers and students focus on the logic of the algorithm without worrying about the syntax of a specific programming language.

Why Use Pseudocode?

Using pseudocode has several benefits:

- **Clarity:** It helps in understanding the logic of the algorithm without worrying about syntax.
- **Planning:** It allows programmers to outline their thoughts and plan the steps of the algorithm.
- **Communication:** It is a universal way to convey the steps of an algorithm, making it easier to discuss with others.

Example-1

Determining whether a number is even or odd is a fundamental task in programming and computer science. An even number is divisible by 2 with remainder of zero, whereas an odd number has a remainder of 1 when divided by 2. Below is the pseudocode for this process, followed by an explanation.

Algorithm 1 Pseudocode for determining if a number is even or odd.

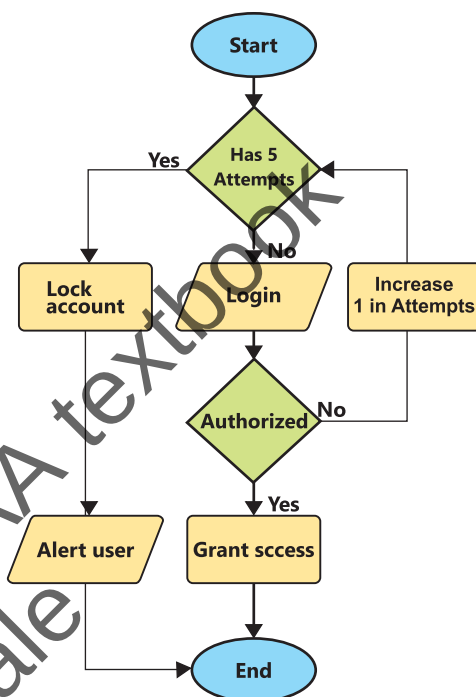


Figure 4.5: Flowchart for a login system



```
1:      Procedure CheckEvenOdd(number)
2:      Input: No. {The number to be checked}
3:      Output: "Even" if number is even, "Odd" if No. is odd
4:      Begin
5:      if (No. % 2 == 0) then
6:          print "No is Even"
7:      else
8:          print "No. is Odd"
9:      End if
10:     End
```

Example-2

Determining whether a number is prime is a fundamental task in number theory and computer science. A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself. Below is the pseudocode for this process, followed by an explanation.

Algorithm 2 Pseudocode for determining if a number is prime.

```
1:      Procedure Is Prime(number)
2:      Input: number {The number to be checked}
3:      Output: True if number is prime, False otherwise
4:      Begin
5:      if (number <= 1) then
6:          return False
7:      end if
8:      for I from 2 to sqrt(number) do
9:          if (number % i == 0) then
10:             return False
11:          end if
12:      end for
13:      return True
14:     End
```

Did You Know?

Pseudocode is often used in software development before writing the actual code to ensure that the logic is sound and to facilitate communication between team members who may be using different programming languages.



Class ACTiViTy



Create Your Own Pseudocode: Divide the students into small groups and assign each group a different simple problem, such as finding the maximum number in a list or calculating the factorial of a number. Ask them to write the pseudocode for their assigned problem and then present it to the class.

1. Differentiating Flowcharts and Pseudocode

Flowcharts and pseudocode are both tools used to describe algorithms, but they do so in different ways. Understanding their differences can help you decide which method is more suitable to use for your scenario.

Pseudocode	Flowcharts
• Pseudocode uses plain language and structured format to describe the steps of an algorithm. • It is read like a story, with each step is written out sequentially. • Pseudocode communicates the steps in a detailed, narrative-like format. • It is particularly useful for documenting algorithms in a way that can be easily converted into actual code in any programming language.	• Flowcharts use graphical symbols and arrows to represent the flow of an algorithm. • It is like watching a movie, where each symbol (such as rectangles, diamonds, and ovals) represents a different type of action or decision, and arrows indicate the connection and direction of the flow. • Flowchart communicates the process in a visual format, which can be more effective for understanding the overall flow and structure. • They are useful for identifying the steps and decisions in an algorithm at a glance.

Table 4.2 Difference between Pseudocode and Flowcharts

Algorithm 3 presents the pseudocode for checking a valid username and password.



1. **Procedure** CheckCredentials(username, password)
2. **Input:** username, password
3. **Output:** Validity message
4. **Begin**
5. validUsername = "user123" {Replace with the actual valid username}
6. validPassword = "pass123" {Replace with the actual valid password}
7. if (username == validUsername) then
 - 8: if (password == validPassword) then
 - 9: print "Login successful"
 - 10: else
 - 11: print "Invalid password"
 - 12: end if
 - 13: else
 - 14: print "Invalid username"
 - 15: end if
 - 16: End

4.4 Evaluation Techniques for an Algorithm

Techniques to essential algorithms are essential to understand how efficiently they solve problems. In this section, we will explore different techniques for evaluating algorithms, focusing on their time and space complexities.

Time Complexity

Time Complexity measures how fast or slow an algorithm performs. It shows how the running time of an algorithm changes as the size of the input increases. Here's an easy way to understand it:

Imagine you have a list of names, and you want to find a specific name. If you have 10 names, it might only take a few seconds to look through the list. But what if you have 100 names? Or 1,000 names? The time it takes to find the name increases as the list gets longer. Time complexity helps us understand this increase.

Did You Know?

Time complexity is usually expressed using Big O notation, like $O(n)$, $O(\log n)$, or $O(n^2)$. It helps us compare different algorithms to see which one is faster!



Class ACTiViTy



Think of a simple task, like finding the largest number in a list. Write down the steps you would take to complete this task. Now, imagine the list has 10 numbers, then 100 numbers. How do the steps change?

Space Complexity

Space complexity measures the amount of memory an algorithm uses relative to input size. It is essential to consider both the memory required for the input and any extra memory used by the algorithm.

Designing and evaluating algorithms involves activities like dry runs and simulations to ensure they work as intended.

4.5 Dry Run

A dry run involves manually going through the algorithm with sample data to identify any errors.

Dry Run of a Flowchart

A dry run of a flowchart involves manually walking through the flowchart step-by-step to understand how the algorithm works without using a computer. This helps identify any logical errors and understand the flow of control.

Example: Calculating the Sum of Two Numbers

Consider the flowchart given in figure 7.7 for adding two numbers:

Steps to dry run this flowchart:

1. Start
2. Input the first number (e.g., 3)
3. Input the second number (e.g., 5)
4. Add the two numbers ($3 + 5 = 8$)
5. Output the result (8)
6. Stop

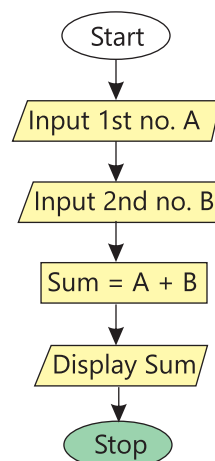


Figure 4.6: Flowchart for adding two numbers



Class ACTiViTy

Dry Run a Flowchart

Draw a flowchart for finding the largest of two numbers. Perform a dry run for the numbers 7 and 4. Write down each step and the values of variables.

Dry Run of Pseudocode

A dry run of pseudocode involves manually simulating the execution of the pseudocode line-by-line.

This helps in verifying the logic and correctness of the algorithm.

Example: Finding the Maximum of Two Numbers

Consider the pseudocode for finding the maximum of two numbers:

Algorithm 4 FindMax

1. Input: num1, num2
2. if num1 > num2 then
3. max = num1
4. else
5. max = num2
6. end if
7. Output: max

Steps to dry run this pseudocode:

1. Input num1 and num2 (e.g., 10 and 15)
2. Check if num1 > num2 (10 > 15: False)
3. Since the condition is False, max = num2 (max = 15)
4. Output max (15)

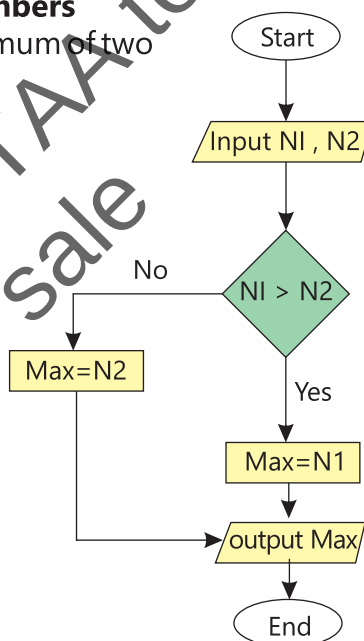


Figure 4.7: Flowchart for finding maximum of two numbers

Did You Know?

- Dry running your code or algorithm helps catching errors early in the development process, saving time and effort.
- Many professional programmers and computer scientists use dry running as a debugging technique to ensure their algorithms work correctly!



Simulation

Simulation is the use of computer programs to create a model of a real-world process or system. This helps us understand how things work by testing different ideas or algorithms without needing to try them out in real life.

Why Use Simulation?

- 1. Testing Algorithms:** We can use simulation to see how well an algorithm works with different types of data. For example, if we want to test a new way to sort numbers, we can simulate it with different sets of numbers to see how fast it is.
- 2. Exploring Scenarios:** Simulation allows us to create many different situations to see what happens. For example, in a science experiment about plant growth, we can simulate different amounts of water or sunlight to find out which conditions help plants grow best.

Benefits of Simulation

- **Cost-Effective:** It is often cheaper and faster to run simulations than to conduct real experiments.
- **Safe:** We can test dangerous situations, like a fire in a building, without putting anyone at risk.
- **Repeatable:** We can run the same simulation multiple times with different settings to observe how things change.

Examples of Simulation

- 1. Weather Forecasting:** Meteorologists use simulations to predict the weather. They input data about temperature, humidity, and wind speed into a computer model to see how the weather might change over the next few days.

Class ACTIVITY



Simulation Game

Objective: To experience managing a system and making decisions to keep it functional.

Required Material: Computers or tablets with internet access, city simulation game (e.g., SimCity).

Activity Type: Pair (Group of two students)

Activity Tasks Detail: Begin with an introduction to the simulation game, explaining its objectives and mechanics. Students will then play the game in pairs, making strategic decisions to manage their city. After gameplay, a debriefing session will allow students to discuss their experiences, challenges, and strategies. The teacher will link these experiences to system management concepts discussed in class.

Output: Hands-on gameplay experience and reflection on system management challenges and strategies.



- 2. Traffic Flow:** City planners can simulate traffic to see how changes to roads or traffic lights might affect the flow of cars. This helps them design better roads and reduce traffic jams.

4.6 Introduction to LARP (Logic of Algorithms for Resolution of Problems)

LARP stands for Logic of Algorithms for resolution of Problems. It is a fun and interactive way to learn how algorithms work by actually running them and seeing the results. Think of it as a playground where you can experiment with different algorithms and understand how they process data.

Why is LARP Important?

LARP helps you:

- Understand how algorithms work. For instance, refer to Figure 4.8, which illustrates an algorithm designed to determine the applicability of tax on the annual salary of a person.
- See the effect of different inputs on the output.
- Practice writing and improving your own algorithms.

Writing Algorithms

Writing algorithms using LARP involves a structured and simplified approach to developing logical solutions for computational problems. LARP employs a clear syntax that begins with a START command and ends with an END command, ensuring that each step of the algorithm is easy to follow. Within this framework, instructions are provided in a straightforward manner, such as using WRITE to display messages, READ to input values, and conditional statements like IF...THEN...ELSE to handle decision-making processes.

Here's an example of a simple algorithm to check if a number is even or odd:

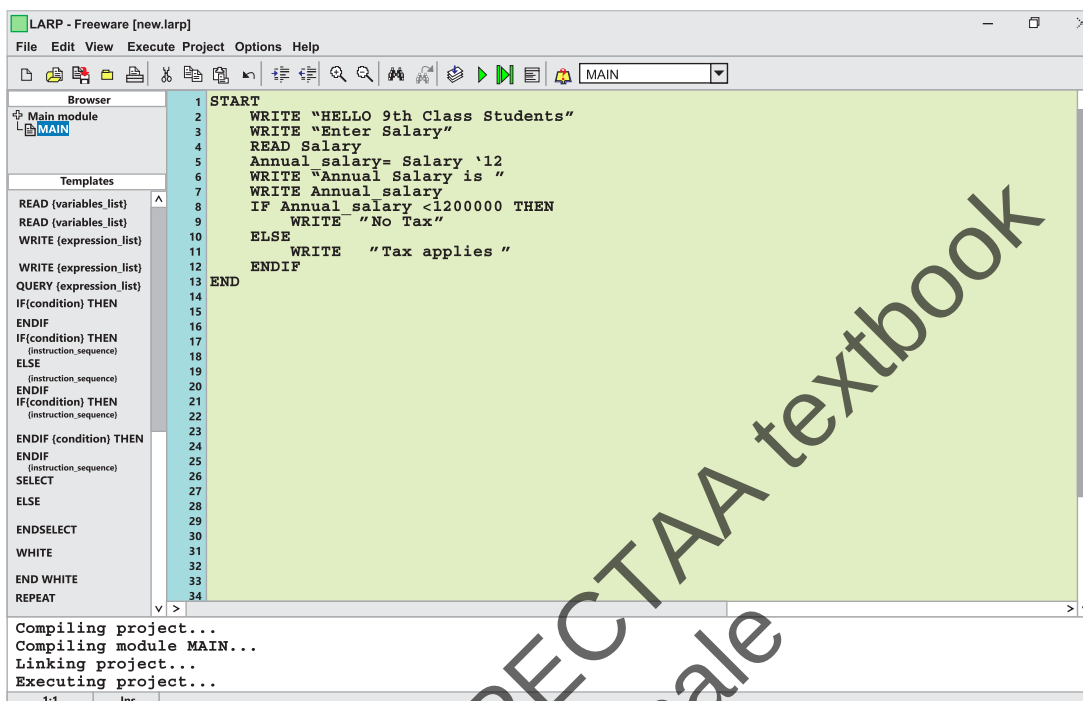


Figure 4.8: LARP Software

```
START
  WRITE "Enter a number"
  READ number
  IF number % 2 == 0 THEN
    WRITE "The number is even"
  ELSE
    WRITE "The number is odd"
  ENDIF
END
```

Drawing Flowcharts in LARP

Drawing flowcharts in LARP involves visually representing the algorithm's steps using standard flowchart symbols such as rectangles for processes, diamonds for decisions, and parallelograms for input/output operations. Once the flowchart is created, it can be executed in LARP by translating the flowchart into LARP syntax, which uses straightforward commands like START, WRITE, READ, IF...THEN...ELSE, and END. This process allows students to visualize the logic of their algorithm



and see its step-by-step execution. For example, Figure 4.9 shows a flowchart for determining whether a student's grade is above 'A' or not. We can execute the flowchart to verify its correctness. This hands-on approach reinforces understanding of how a flowchart works.

4.7 Error Identification and Debugging

When we write algorithms or create flowcharts in LARP, we sometimes make mistakes called errors or bugs. These mistakes can prevent our algorithms from functioning correctly. Error handling and debugging are processes that help us find and fix these errors.

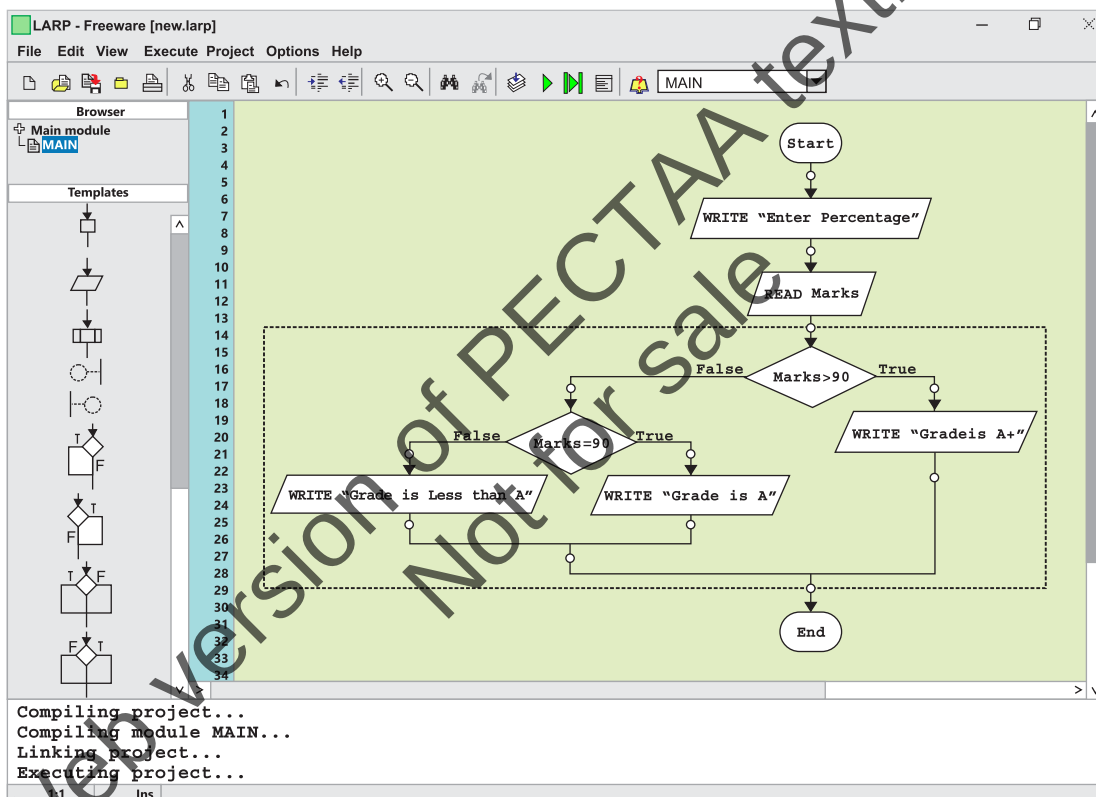


Figure 4.9 Flowchart in LARP

Types of Errors

There are three main types of errors you might encounter:

- **Syntax Errors:** These occur when we write something incorrectly in our algorithm or flowchart. For example, missing a step or using the wrong symbol.
- **Runtime Errors:** These happen when the algorithm or flowchart is being



executed. For example, trying to perform an impossible operation, such as dividing by zero.

- **Logical Errors:** These are mistakes in the logic of the algorithm that cause it to behave incorrectly. For example, using the wrong condition in a decision step.

Did You Know?

Syntax errors are the easiest to find because the LARP tool usually points them out. However, logical errors are the hardest to find because the algorithm still runs but does not produce correct answers.



TidBiTs

Always read error messages carefully. They often tell you exactly where the problem is.

Common Error Messages in LARP

Here are some common error messages you might see in LARP and what they mean:

- **Missing Step** - You probably forgot to include an important step in your algorithm.
- **Undefined Variable** - You are using a variable that hasn't been defined yet.
- **Invalid Operation** - You are trying to perform an operation that is not allowed, like dividing by zero.

Class ACTiViTy



Create a simple flowchart in LARP that calculates the average of three numbers. Introduce a syntax error, a runtime error, and a logical error in your flowchart. Then, try to fix them using the debugging techniques we discussed.



Did You Know?

The term "debugging" comes from an actual bug—a moth—that was found causing problems in an early computer. The moth was removed, and the process was called "debugging".

Summary

- Computational thinking is important skill that enables individuals to solve complex problems using methods that mirror the processes involved in computer science.
- An algorithm is a step-by-step set of instructions to solve a problem or complete a task.
- Understanding the problem is the first and most important step in problem-solving, especially in computational thinking.
- Simplifying a problem involves breaking it down into smaller, more manageable sub-problems.
- Choosing the best solution involves evaluating different approaches and selecting the most efficient one.
- Flowcharts are visual representations of the steps in a process or system, depicted using different symbols connected by arrows.
- Pseudocode is a way of representing an algorithm using simple and informal language that is easy to understand. It combines the structure of programming languages with the readability of plain English, making it a useful tool for planning and explaining algorithms.
- Time Complexity is a way to measure how fast or slow an algorithm performs. It tells us how the running time of an algorithm changes as the size of the input increases.
- Space complexity measures the amount of memory an algorithm uses in relation to the input size. It is important to consider both the memory needed for the input and any additional memory used by the algorithm.



EXERCISE

Multiple Choice Questions

1. Which of the following best defines computational thinking?

- a) A method of solving problems using mathematical calculations only.
- b) A problem-solving approach that employs systematic, algorithmic, and logical thinking.
- c) A technique used exclusively in computer programming.
- d) An approach that ignores real-world applications.

2. Why is problem decomposition important in computational thinking?

- a) It simplifies problems by breaking them down into smaller, more manageable parts.
- b) It complicates problems by adding more details.
- c) It eliminates the need for solving the problem.
- d) It is only useful for simple problems.

3. Pattern recognition involves:

- a) Finding and using similarities within problems
- b) Ignoring repetitive elements
- c) Breaking problems into smaller pieces
- d) Writing detailed algorithms

4. Which term refers to the process of ignoring the details to focus on the main idea?

- a) Decomposition
- b) Pattern recognition
- c) Abstraction
- d) Algorithm design

5. Which of the following is a principle of computational thinking?



- a) Ignoring problem understanding b) Problem simplification
- c) Avoiding solution design d) Implementing random solutions

6. Algorithms are:

- a) Lists of data
- b) Graphical representations
- c) Step-by-step instructions for solving a problem
- d) Repetitive patterns

7. Which of the following is the first step in problem solving according to computational thinking?

- a) Writing the solution b) Understanding the problem
- c) Designing a flowchart d) Selecting a solution

8. Flowcharts are used to:

- a) Code a program b) Represent algorithms graphically
- c) Solve mathematical equations d) Identify patterns

9. Pseudocode is:

- a) A type of flowchart
- b) A high-level description of an algorithm using plain language
- c) A programming language
- d) A debugging tool

10. Dry running a flowchart involves:

- a) Writing the code in a programming language
- b) Testing the flowchart with sample data
- c) Converting the flowchart into pseudocode
- d) Ignoring the flowchart details

Short Questions

1. Define computational thinking.
2. What is decomposition in computational thinking?
3. Describe abstraction in problem-solving.
4. What is an algorithm?
5. How does problem understanding help in computational thinking?



6. Why we use flow charts?
7. What is purpose of pseudocode.
8. How do you differentiate between flowcharts and pseudocode?

Long Questions

1. Define computational thinking and explain its significance in modern problem-solving.
2. Define flow chart and explain its different symbols.
3. Explain pattern recognition in the context of computational thinking. How does identifying patterns help in problem-solving?
4. Describe what an algorithm is and give an example.
5. What is LARP? Discuss its importance in learning and practicing algorithms.

Web version of PECTAA textbook
Not for sale



UNIT

5

Web Development with HTML, CSS, and JavaScript



Student Learning Outcomes

By the end of this chapter, you will be able to:

- Understand JavaScript syntax and data types.
- Work with variables, operators, and functions in JavaScript.
- Create simple programs using JavaScript.
- Create HTML forms and style them.
- Use JavaScript to handle del events operators variables, and functions.
- Develop static web pages.
- Apply HTML tags appropriately to create web pages.
- Create a basic HTML page.
- Add text, images, and links to a page.
- Create lists and tables.
- Apply styles to HTML elements.
- Work with fonts, colors, and backgrounds.
- Create web pages to display data in the paragraphs and lists.
- Familiarize students with CSS syntax.
- Create layouts with CSS.
- Organize images and text effectively.

Introduction

In this chapter, the fundamentals of web development, Hyper Text Markup Language (HTML), JavaScript and Cascading Style Sheet (CSS) will be explained. By the end of this chapter, you will be able to understand JavaScript syntax and data types, work with variables, operators, and functions, handle events and user inputs, create simple programs, and develop static web pages using HTML and CSS.

5.1 Web Development

Process of creating websites and web applications is called Web development. It means using various programming languages and tools to design, build, and



maintain websites.

Why Learn Web Development?

Web development is a valuable skill for several reasons:

- **Digital Literacy:** When you learn web development, you find out how websites are made. You learn about HTML, which is like the skeleton of a web page, CSS, which makes the web page look nice, and JavaScript, which makes the web page interactive. This helps you understand how the internet works.
- **Career Opportunities:** Opens up a wide range of job prospects in the growing IT industry. Web developers can get many different kinds of jobs. You can become a web developer, web designer, and more. Many companies need web developers to create and maintain their websites. This means you can find good jobs in many places.
- **Problem-Solving:** When you build a website, you solve many problems. For example, if a website is slow, you figure out why and fix it. This helps you think logically and solve problems better.
- **Creativity:** Allows you to create visually appealing and interactive websites. Web development lets you be creative. You can design websites with cool layouts, colors, and interactive features. For example, you can create a personal blog or a portfolio to show your artwork, making your own unique website.
- **Entrepreneurship:** With web development skills, you can start your own online business. For example, if you make crafts, you can build a website to sell them. Or, you can create a new web service, like a fun app, and share it with the world.

Did You Know?

Hotmail, one of the first web-based email services, was created by students Sabeer Bhatia and Jack Smith while they were at Stanford University. It was later acquired by Microsoft for \$400 million.

5.2 Basic Components of Web Development

Web development involves creating websites and web applications. It has three main components:

1. **Front-end Development:** This focuses on what users see and interact with on a website. The following fundamentals are used to design interactive Front-ends:
 - **HTML** structures the content on web pages, like headings, paragraphs, images, and links.



- **CSS** styles the content on web pages, changing colors, fonts, and layout to enhance the appearance.
 - **JavaScript** adds interactivity to web pages, making them dynamic and engaging. It allows features such as forms, animations, and games.
2. **Back-end Development:** This manages the behind-the-scenes functionality of a website, including servers, databases, and application logic. Key back-end technologies are:
- **Web Servers** are computers that store and deliver web pages to users when they enter a URL.
 - **Databases** store and manage data, like user information, product details, and website content.
 - **Back-end Programming Languages** like PHP, Python, and Ruby handle tasks such as processing forms, and managing user logins.

Did You Know?

The first website was created by Tim Berners-Lee in 1991 and it is still accessible at <http://info.cern.ch>. It was a simple page with links to information about the World Wide Web project.

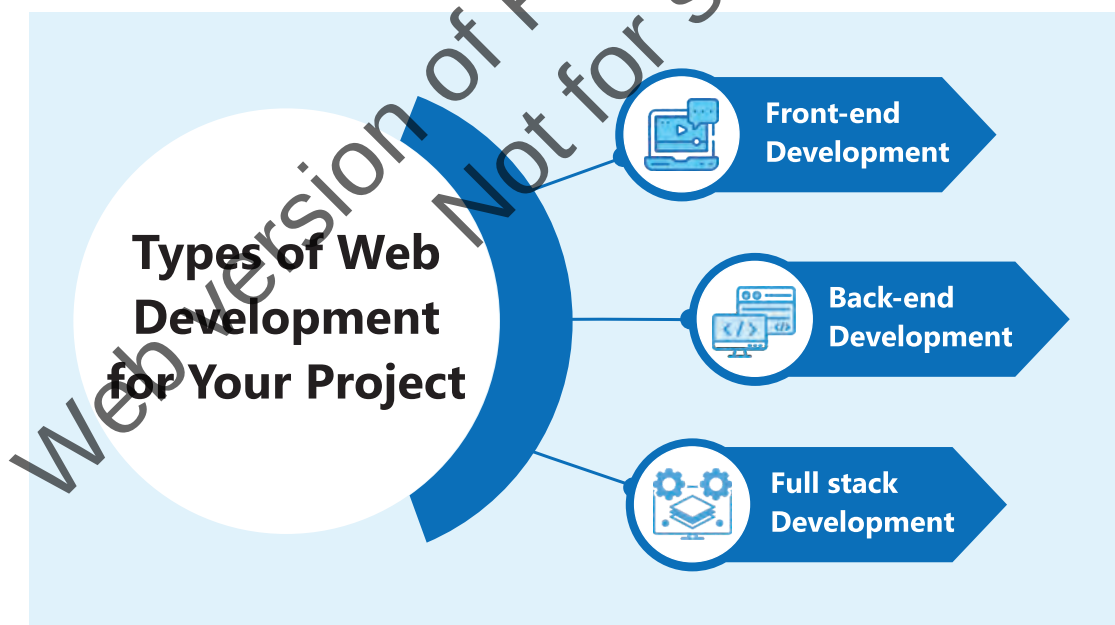


Figure 5.1: Types of Web Development



Example: Login System

A login system is a common feature in web development, allowing users to access their accounts on a website. This example will illustrate the roles of front-end and back-end development, as well as the concept of full-stack development.

3. Full-Stack Development

In the case of login system, a full-stack developer will create the User Interface (UI) for front-end and handle user authentication and database interaction for back-end.

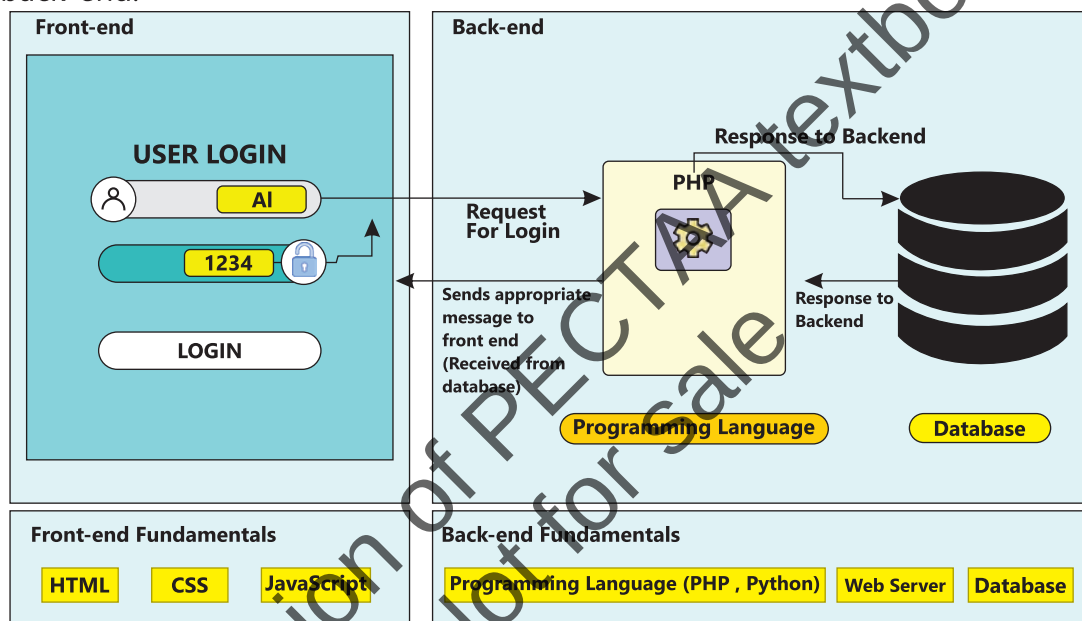


Figure 5.2: Graphical abstract of login system

Did You Know?

Full-stack developers are in high demand because they can manage and develop all aspects of a web application, making them versatile and valuable in the tech industry.

5.3 Getting Started with HTML

HTML is the standard language used to create web pages. Think of HTML as the building blocks of a website. Just like LEGO pieces (as shown in Figure 5.3) come together to build a structure, HTML tags come together to build a web page.



Setting up a Development Environment

To start creating websites, you need a few basic tools and environments:

- **Text Editor:** This is where you write your HTML code. Popular text editors include Notepad++, Sublime Text, and Visual Studio Code.
- **Web Browser:** You will use this to view and test your HTML files. Common web browsers are Google Chrome, Mozilla Firefox, and Microsoft Edge.

Did You Know?

Start with a simple text editor and a web browser. Once you are comfortable with HTML, you can explore more advanced tools.

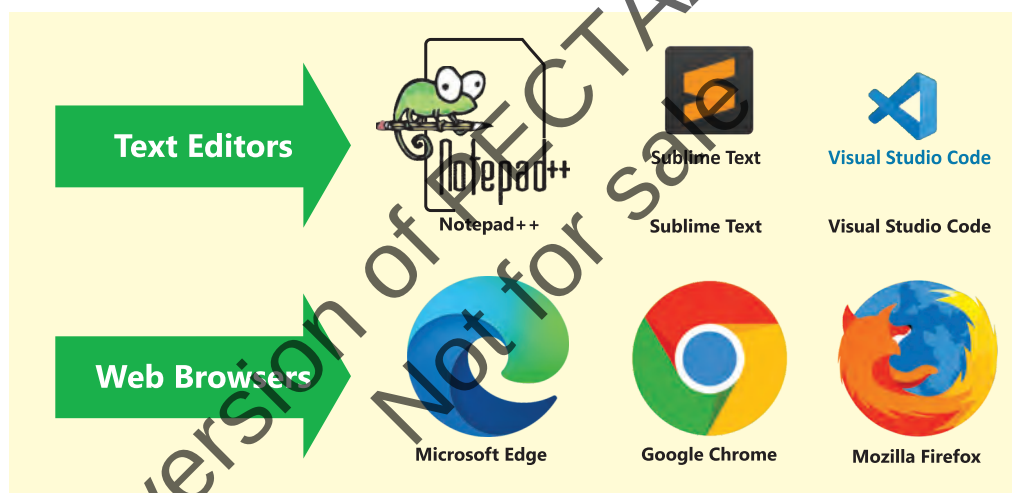


Figure 5.3: Fundamentals of website development environment

Creating a "Hello, World!" HTML Application

To create a basic HTML application that displays "a message" on a web page, follow these simple steps:

Writing the HTML Code

1. **Open your text editor,** You can use Notepad, Notepad++, Sublime Text, or any other text editor.
2. **Write the following HTML code** into your text editor.
3. **Save your file** with a html extension, for example, My_first_website.html.



HTML Code Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>My First Web Page</title>
  </head>
  <body>
    <h1>Welcome to My Website</h1>
    <p>This is my first web page. I am learning HTML in the 9th
class!</p>
  </body>
</html>
```

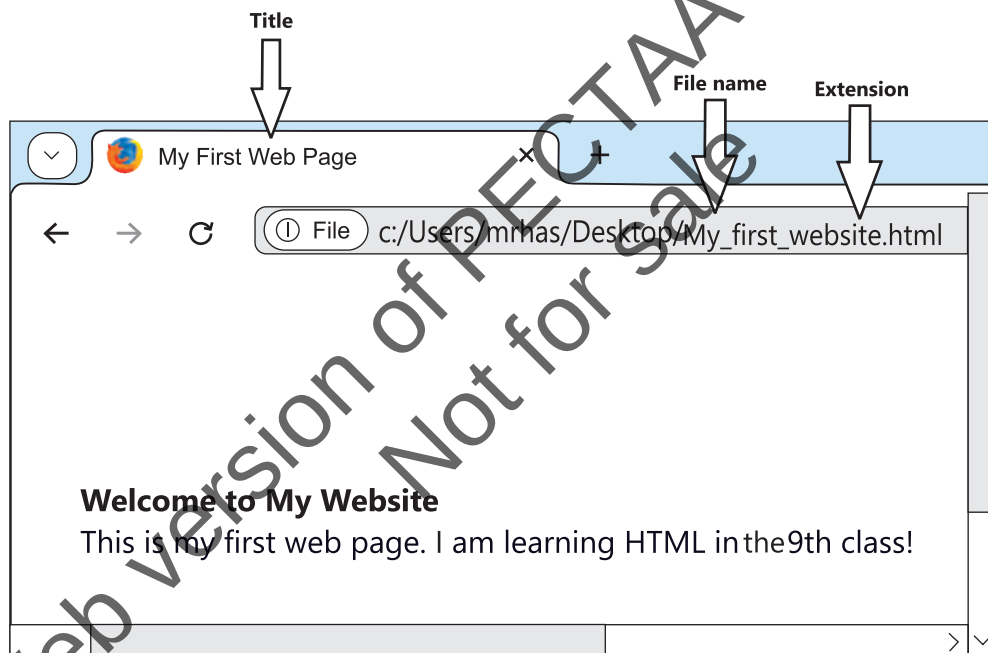


Figure 5.4: HTML in Web Browser

Viewing the HTML File

1. Open Your Web browser (Google Chrome, Mozilla Firefox, and others).
2. Double-click on your file named My_first_website.html.
3. You should see the text welcome to my website displayed on the web page as shown in Figure 5.4.



TidBiTs

When saving your HTML files, always remember to add the .html extension at the end of your file name. This helps your computer recognize it as a web page. If you are using Notepad, make sure to select "All Files" in the "Save as type" dropdown, then type your file name followed by .html (e.g., hello_world.html).

Did You Know?

If you make any changes to the HTML file, refresh the web page in your browser to see the updated content.

5.4 HTML Basic Structure

A structured HTML document is easier to read and understand. Properly nested and well-organized elements help developers and browsers interpret the content correctly. This organization ensures that the web page displays as intended.

Every HTML document has a basic structure where:

- `<!DOCTYPE html>`: This line tells the browser that this is an HTML5 document.
- `<html>`: This is the root element of an HTML page.
- `<head>`: This section contains meta-information about the HTML document, like the title.
- `<title>`: This sets the title of the web page, which appears in the browser tab.
- `<body>`: This section contains the content of the web page that you see in the browser.
- `<h1>`: This defines a large heading.
- `<p>`: This defines a paragraph.

HTML Tags

Elements that make up an HTML document are called tags. A web page's structure and content are defined by them. On the basis of structure, HTML Tags are categorized into two types:

1. **Paired Tags:** Comes in pairs an opening Tag and closing Tag i.e `<p> .....</p>`.
2. **Singular Tags:** Do not need closing Tags. They are also known as self-closing Tags i.e., `<img>`, `<br>`.



5.5 Creating Content with HTML

Content in HTML is the main information on a web page that users read and interact with. It includes text, images, videos, links, and other elements that convey the purpose and message of the page. This makes it easier for people to find your site.

Headings

Headings in HTML, ranging from `<h1>` to `<h6>`, are used to define the structure and hierarchy of content on a web page. Here's why they are important:

Importance of Headings

1. **Organizing Content Headings** helps organize the content into sections and subsections, making it easier for users to read and understand. `<h1>` is typically used for the main title of the page, while `<h2>` to `<h6>` are used for subheadings in decreasing order of importance.
2. **Search Engine Optimization (SEO):** Search engines use headings to understand the structure and main topics of a web page. Proper use of headings can improve the page's SEO, helping it rank higher in search results.
3. **Consistent Formatting:** Using standard heading tags ensures consistent formatting across different browsers and devices.

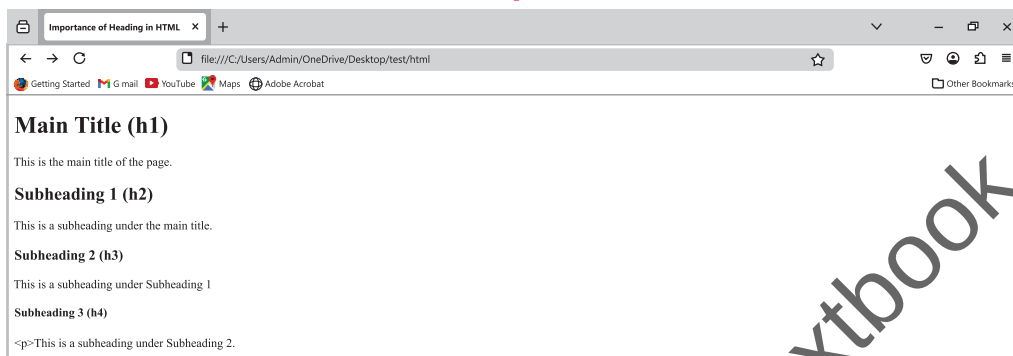
Example

Here is an example of how different heading levels can be used to organize content in a hierarchical structure:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Importance of Headings in HTML</title>
  </head>
  <body>
    <h1>Main Title (h1)</h1>
    <p>This is the main title of the page.</p>
    <h2>Subheading 1 (h2)</h2>
    <p>This is a subheading under the main title.</p>
    <h3>Subheading 2 (h3)</h3>
    <p>This is a subheading under Subheading 1.</p>
  </body>
</html>
```



Output



Paragraphs

Paragraphs in HTML are used to organize and separate text into readable sections. Each paragraph creates a block of text with space above and below it, making the content easier to read. Paragraphs start with the `<p>` tag and it ends with `</p>`.

Links

Links in HTML are used to connect one web page to another. They allow you to click on words or images to go to different parts of the same web page or to other web pages on the internet.

Links are created using the `<a>` tag.

```
<a href="http://www.example.com">Visit Example.com</a>
```

```
<a href="mailto:example@example.com">Send Email</a>
```

Sometimes, links in HTML can also let you click to send an email. These special links start with `mailto:` and when you click them, they open your email program so you can send a message to the email address in the link.

Images

Images are important in HTML because they make web pages more attractive and engaging. Additionally, using images helps with branding, as logos and specific visuals make it easier for users to recognize a brand. Lastly, including alternate text for images ensures that visually impaired users can understand what the images represent.

Images are added using the `<img>` tag.

```

```

Lists

Lists improve readability by breaking complex ideas into simpler parts, allowing users to scan for details easily. Overall, lists make the content more organized and



accessible for everyone. You can create ordered (numbered) and unordered (bulleted) lists.

Unordered List

`<ul>`

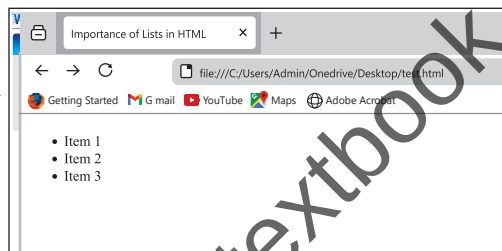
`<li>Item 1</li>`

`<li>Item 2</li>`

`<li>Item 3</li>`

`</ul>`

Result



Output can be given side by

Ordered List

`<ol>`

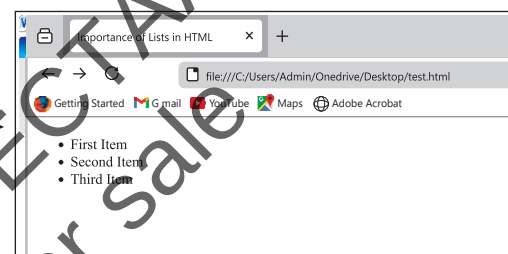
`<li>First item</li>`

`<li>Second item</li>`

`<li>3rd item</li>`

`</ol>`

Result



Creating Tables in HTML

Tables in HTML are used to display data in a structured format, allowing for easy comparison and organization of information. A table is created using the `<table>` tag, which contains rows defined by `<tr>` (table row) tags, and each row consists of cells represented by `<td>` (table data) tags. Additionally, headings for the table can be added using `<th>` (table header) tags to provide context for the data.

Example:

`<table>`

`<tr>`

`<th>Name</th>`

`<th>Age</th>`

`</tr>`

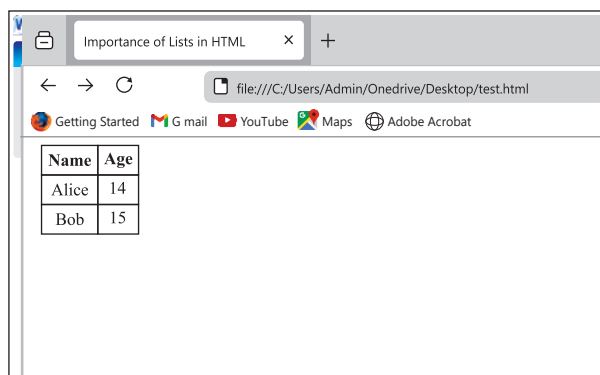
`<tr>`

`<td>Alice</td>`

`<td>14</td>`

`</tr>`

Output





```
<tr>  
    <td>Bob</td>  
    <td>15</td>
```

```
</tr>
```

```
</table>
```

HTML Comments

In HTML, comments can be extremely useful for:

- Explaining the purpose of a specific section of code
- Leaving reminders for future edits
- Temporarily disabling code for testing purposes

Syntax of HTML Comments

HTML comments begin with `<!--` and end with `-->`. Any text placed within these markers will be treated as a comment and will not be rendered by the browser.

```
<!-- This is a comment -->
```

5.6 Styling with CSS

Styling with Cascading Style Sheets (CSS) is very important for improving the visual appearance of webpages and improving user experience. CSS allows web developers to control the colors, fonts, layout, and overall design of HTML elements, separating the content from the presentation. CSS offers various properties and selectors to apply styles to specific elements, enabling responsive design that automatically adjusts to different screen sizes and devices.

Basic Structure of CSS

The basic structure of CSS is essential for applying styles to HTML elements effectively. CSS is composed of rules that consist of selectors and declarations. Selectors specify which HTML elements the styles will apply to, while declarations define the specific styles to be applied, including properties and their corresponding values. A typical CSS rule follows this format:

For example, A simple rule of CSS can change the color & size of all heading in a web page.

```
selector { property: value;
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {
```

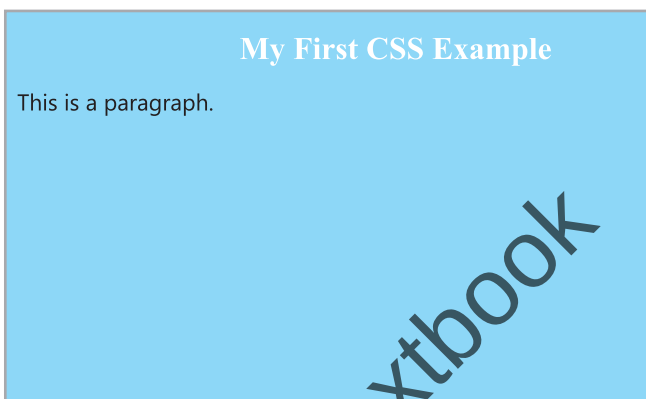
```
    background-color: lightblue;
```



```

}
h1{
  color: white;
  text-align: center;
}
p{
  font-family: verdana;
  font-size: 20px;
}
</style>
</head>
<body>
<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>
</body>
</html>

```



Integrating CSS in HTML

Integrating CSS with HTML is essential for styling web pages and it can be done in three primary ways: inline, internal, and external styles.

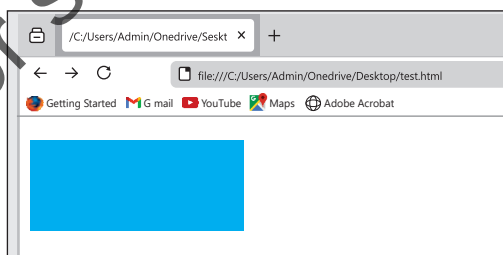
- 1. Inline Styles:** This method involves adding CSS directly to individual HTML elements using the style attribute. For example, `<h1 style="color: blue;">Hello World</h1>` changes the color of the heading to blue. While easy for quick changes, inline styles can make the code cluttered and less maintainable.
- 2. Internal Styles:** CSS can also be included in the `<head>` section of an HTML document using the `<style>` tag. This method allows you to define styles for the entire page without affecting others.

For instance:

```

<style> h1{
color: yellow;
}

```





</style>

3. External Styles: The most efficient method for larger projects is to use an external CSS file, which is linked to the HTML document with the <link> tag in the <head> section. This keeps the HTML clean and allows for easy updates across multiple pages.

For example:

```
<link rel="stylesheet" href="styles.css">
```

Did You Know?

You can use different fonts by specifying their names in CSS. For example, you can set the font to Arial or Times New Roman.

By integrating CSS in these ways, developers can create visually appealing and well-organized web pages that enhance user experience.

Styling HTML Elements with Fonts, Colors, Backgrounds Styling Fonts

You can change the appearance of text on a web page using CSS. This includes changing the font family, size, weight, and style.

Example of Styling Fonts

Here's how you can style the font of a paragraph:

```
<html>
```

```
<head>
```

```
<style>
```

```
P{
```

```
font-family: Arial, sans-serif;
```

```
font-size: 16px;
```

```
font-weight: bold;
```

```
font-style: italic;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>CSS font-family</h1>
```

```
<p class="P">This is a paragraph, shown in the Arial, sans-serif font font size is 16, font weight is bold and style is italic.</p>
```

```
</body>
```

CSS font-family

This is a paragraph, shown in the Arial, sans-serif font font size is 16, font weight is bold and style is italic.



</html>

In this example:

- font-family: Arial, sans-serif; sets the font to Arial. If Arial is not available, it will use a generic sans-serif font.
- font-size: 16px; sets the font size to 16 pixels.
- font-weight: bold; makes the text bold.
- font-style: italic; makes the text italic.

Creating Layouts and Organizing Content

Creating layouts and organizing content on a web page is an important part of web design. CSS helps you arrange different parts of your web page in an organized way. Here are some basic methods to create layouts and organize content:

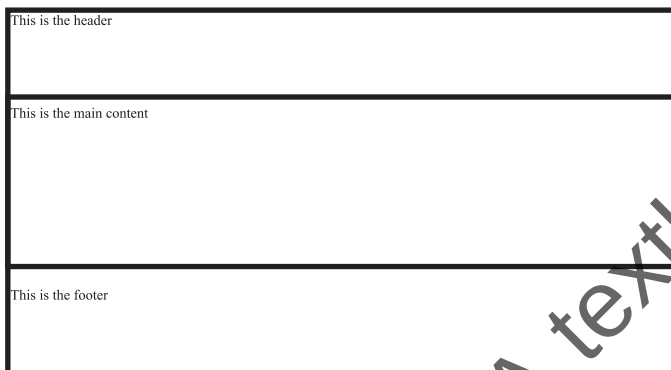
• **Divs and Sections:** HTML elements like <div> and <section> are used to group content together. You can then use CSS to style and position them.

For example:

```
<html>
<head>
<style>
.container{
border: 2px solid;
}
section.header{
height: 100px;
border: 2px solid;
}
section.content{
height: 200px;
border: 2px solid;
}
section.footer{
height: 100px;
border: 2px solid;
}
</style>
</head>
<body>
<div class="container">
<section class="header">This is the header</section>
```



```
<section class="content">This is the main content</section>
<section class="footer">This is the footer</section>
</div>
</body>
</html>
```



• **CSS Grid:** The CSS Grid Layout is a powerful tool for creating complex layouts. It allows you to arrange items into rows and columns. For example:

```
<html>
<head>
<style>
.container{
display: grid;
grid-template-columns: auto auto auto;
grid-gap: 50px;
border: 2px solid;
}
.header{
padding: 20px;
background-color: lightgrey;
}
.footer{
padding: 20px;
background-color: lightgreen;
}
.content{padding: 20Px;
background-color=white;
}.
</style>
</head>
<body>
```



```

<div class="container">
  <section class="header">This is the header</section>
  <section class="content">This is the main content</section>
  <section class="footer">This is the footer</section>
</div>
</body>

</html>

```

This is the header	This is the main content	This is the footer
--------------------	--------------------------	--------------------

- **CSS Flexbox:** Flexbox is another layout tool that helps in arranging items in a flexible and responsive way. It is useful for aligning items in a row or column.

For example:

```

<html>
<head>
<style>
.flex-container {
  display: flex;
  background-color: pink;
}

.flex-container > div {
  background-color: lightblue;
  margin: 10px;
  padding: 20px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>Create a Flex Container</h1>
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>

```



<p>A Flexible Layout must have a parent element with the display property set to flex.</p>

<p>Direct child elements(s) of the flexible container automatically becomes flexible items.</p>

</body>

</html>

Create a Flex Container



A Flexible Layout must have a parent element with the *display* property set to *flex*.

Direct child elements(s) of the flexible container automatically becomes flexible items.

5.7 Introduction to JavaScript

JavaScript is a programming language that is used to make websites interactive and engaging. It allows developers to create things like animations, games, and responsive features that react when you click buttons or move your mouse. For example, when you see a pop-up message on a web page or when an image changes when you hover over it, that's JavaScript at work. Execution of JavaScript in a flowchart is shown in Figure 8.7.

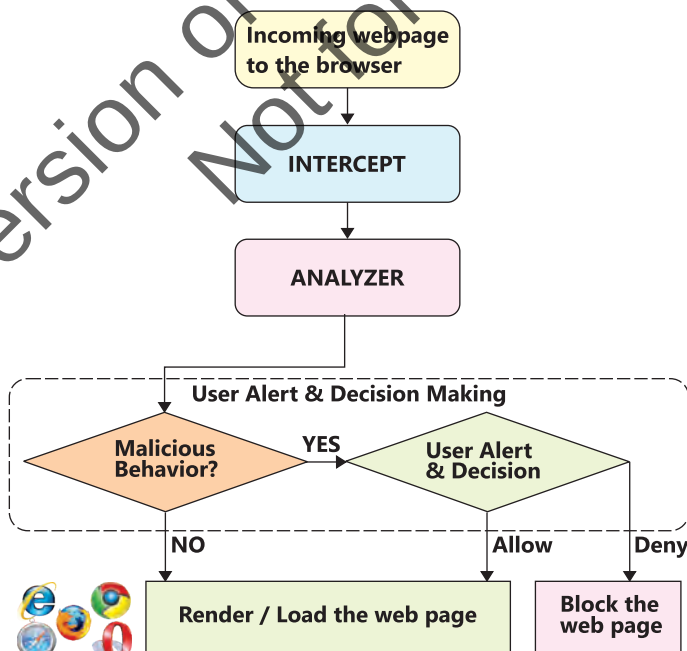


Figure 5.7: JavaScript Execution



Did You Know?

JavaScript was created in just 10 days by Brendan Eich in 1995. It was initially called Mocha, then LiveScript, and finally JavaScript.

Basic Syntax and Examples

Here is a simple example to display an alert message using JavaScript.

```
<!DOCTYPE html>
<html>
  <head>
    <title>JavaScript Example</title>
  </head>
  <body>
    <h1>Welcome to JavaScript</h1>
    <script>
      alert("Hello, its 9th Class ");
    </script>
  </body>
</html>
```

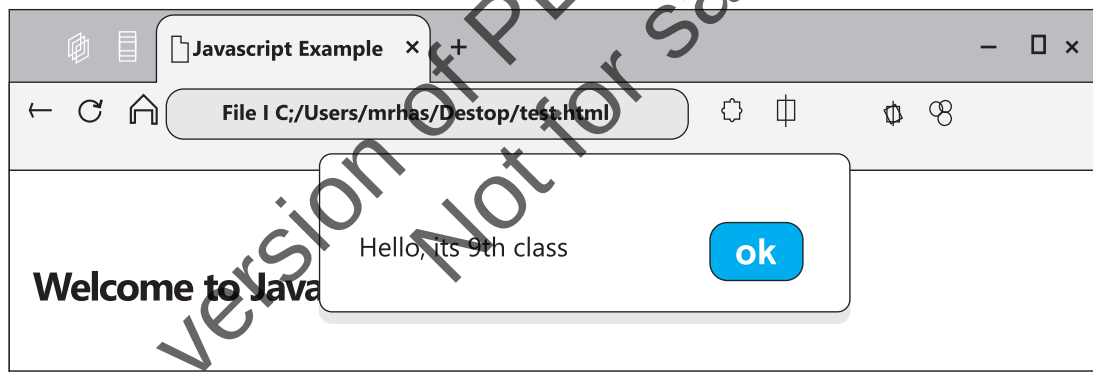


Figure 5.8: Alert Message Example

Class ACTiViTy



Activity: Displaying an Alert

1. Open a text editor (e.g., Notepad).
2. Write the HTML and JavaScript code as shown above.
3. Save the file as "index.html".
4. Open the file in a web browser to see the alert message.



Variables and Data Types

In JavaScript, you can store data using variables. A variable is like a container that holds information which can be used and manipulated in your code.

Declaring Variables

To declare a variable in JavaScript, you use the **var**, **let**, or **const** keyword. Here's an example using **var**:

```
<script>
var name = "Athar";
var age = 15;
alert("Name: " + name + " Age: " + age);
</script>
```

Dry Run Example

Let's dry-run the above script to understand how it works step-by-step:

1. Declare variables: `name = "Athar";` and `age = 15;`
2. Display alert: `alert("Name: " + name + ", Age: " + age);`

This will result in an alert box showing: Name: Athar, Age: 15.

Functions in JavaScript

Functions allow you to reuse code and perform specific tasks. They are like mini-programs that you can run whenever you need them. Let's learn how to create and use functions in JavaScript.

Simple Function

Here's an example of a simple function that displays a greeting message:

```
<script>
function greet() {
  alert("Hello, Student!");
}
greet(); // This calls the function to execute
</script>
```

In this example:

- The function `greet()` declares a function named `greet`.
- `alert("Hello, Student!");` is the code that runs when the function is called.
- `greet();` calls the function, displaying the alert message.

Function with Parameters

Sometimes, you want your function to do something with input values. You can achieve this using parameters.



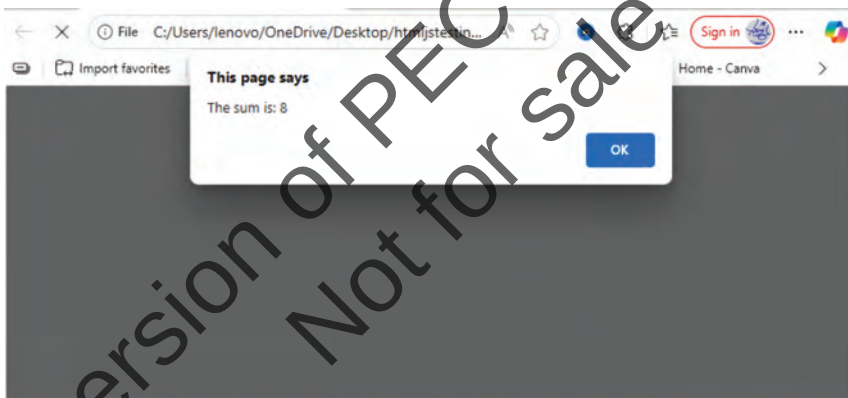
Function with Multiple Parameters

You can also create functions that take multiple parameters. Here's an example:

```
<script>
function addNumbers(a, b) {
  var sum = a + b;
  alert("The sum is: " + sum);
}
addNumbers(5, 3); // This calls the function with the parameters 5 and 3
</script>
```

In this example:

- The function `addNumbers(a, b)` declares a function that takes two parameters, `a` and `b`.
- `var sum = a + b;` calculates the sum of `a` and `b`.
- `alert('The sum is: ' + sum);` displays the result of the addition.
- `addNumbers(5, 3);` calls the function with the arguments 5 and 3, resulting in the alert message "The sum is: 8".



TidBiTs

Parameters are placeholders for values that you pass to the function. You can use these values inside your function.

Did You Know?

Functions can take any number of parameters. You can pass different values each time you call the function.



Class ACTiViTY



Creating a Function

1. Write a function that takes a name as a parameter and displays a personalized greeting. Call the function with your name to see the greeting.
2. Write a function that calculates the area of a rectangle given its length and width.

Summary

- Web development is the process of creating websites and web applications. It involves using various programming languages and tools to design, build, and maintain websites.
- Front-end Development focuses on what users see and interact with on a website.
- Back-end Development manages the behind-the-scenes part of a website, like servers, databases, and application logic.
- HTML stands for HyperText Markup Language. It's the standard language used to create web pages. Think of HTML as the building blocks of a website.
- Content in HTML is the main information on a web page that users read and interact with.
- Headings in HTML, ranging from `<h1>` to `<h6>`, are used to define the structure and hierarchy of content on a web page.
- Paragraphs in HTML are used to organize and separate text into readable sections.
- Links in HTML are used to connect one web page to another. They allow you click on words or images to go to different parts of the same page or to other pages on the internet.
- In HTML, comments are used to insert notes or explanations within the code.
- Styling with CSS (Cascading Style Sheets) is essential for enhancing the visual appearance of web pages and improving user experience.



EXERCISE

Multiple Choice Questions

1. Which of the following tag is not a correct HTML tag?

- (a) `<div>` (b) `<span>` (c) `<head>` (d) `<foot>`

2. What does CSS stand for?

- (a) Cascading Style Sheets (b) Computer Style Sheets
(c) Creative Style Sheets (d) Colorful Style Sheets

3. Which of the following tag is used to create a hyperlink in HTML?

- (a) `<link>` (b) `<a>` (c) `<href>` (d) `<nav>`

4. Which HTML attribute is used to define inline styles?

- (a) `class` (b) `style` (c) `font` (d) `styles`

5. Which of the following is the correct syntax for a CSS rule?

- (a) `selector {property: value;}` (b) `selector: {property=value;}`
(c) `selector {property=value}` (d) `selector: {property: value;}`

6. In HTML, which markup is used for comments?

- (a) `/* */` (b) `/- /` (c) `<--` (d) `<!-- -->`

7. Which HTML tag is used to create an unordered list?

- (a) `<ol>` (b) `<ul>` (c) `<li>` (d) `<list>`

8. Which tag is used to display a horizontal line in HTML?

- (a) `<br>` (b) `<hr>` (c) `<line>` (d) `<hline>`

Short Questions

1. What is the purpose of the `<head>` tag in HTML?
2. Explain the difference between an ordered list and an unordered list in HTML.
3. How do you add a comment in HTML?
4. What are the different ways to apply CSS to an HTML document?
5. How can you include JavaScript in an HTML file?
6. Describe the syntax for creating a hyperlink in HTML.



7. What is the function of the <div> tag in HTML?
8. What is the use of the <table> tag in HTML?
9. Explain the box model in CSS.

Long Questions

1. Explain the process of setting up a development environment for web development. By discussing the necessary softwares and tools.
2. Create a basic HTML page that includes a header, a paragraph, an image, and a hyperlink.
3. Create a table using HTML which contains 3 rows and 2 column?
4. Describe any three CSS selectors and provide examples of each.
5. Explain the process of creating a responsive web page using java script with the help of an examples.

Web version of PECTAN textbook
Not for sale



UNIT 6

Data Science and Data Gathering



Student Learning Outcomes

By the end of this chapter, you will be able to:

- Identify and differentiate between qualitative and quantitative data, and explain their importance in data analysis.
- Organise data effectively and apply basic data analysis techniques to draw meaningful conclusions from both structured and unstructured data.
- Gain knowledge of various data collection methods and tools, including surveys, questionnaires, and online sources, and understand how to use these methods to gather relevant data.
- Describe different data storage techniques such as spreadsheets, databases, data warehouses, and NoSQL, and understand their applications in managing data.
- Learn data pre-processing techniques and apply quantitative and qualitative analysis methods to interpret data and generate insights.
- Understand the role of cloud storage and collaborative tools in data management, and apply best practices for data protection and remote access.
- Be aware of ethical practices related to data privacy, confidentiality, and legal frameworks, and understand the importance of secure data handling.
- Gain an introduction to data science, including its definition, scope, and fundamental principles, and understand the impact of data science on various applications.
- Explain the concept of big data, its characteristics, and its practical applications in domains, such as retail, healthcare, finance, and transportation.
- Learn about various data science tools and techniques, including predictive modelling and graph analytics.



Introduction

This chapter explores various aspects of data management and its impact on our understanding of information. We begin by examining different types of data and effective ways to collect and store it. We shall learn methods for organising and analysing data, using both quantitative and qualitative techniques. We'll also look into data visualisation, which helps us understand complex information through charts and graphs. Next, we will discuss collaborative tools and cloud computing, along with ethical issues involved in working with data.

6.1 Data

Data consists of raw facts collected about things around us that we can process to generate useful information. It can take many forms, such as numbers, words, measurements, observations, or even images and sounds, and may originate from various sources.

6.2 Data Types

Data can be divided into two broad categories namely qualitative and quantitative.

Qualitative Data

Qualitative data refer to categories or labels used to describe the qualities or characteristics of something rather than its quantity. This type of data offer a way to categorize and provide insights into opinions, behaviours, and experiences through descriptions rather than numbers. Key characteristics of qualitative data include non-numeric, descriptive, and categorical attributes.

- **Non-Numeric:** Qualitative data is represented by words, labels, or symbols instead of numbers. It describes attributes rather than quantities. Examples include the names of students in a class (e.g., Ali, Badar, Qasim).
- **Categorical:** Qualitative data can be divided into categories or classes based on their characteristics. Examples include types of fruit.

Quantitative Data

Quantitative data consists of numbers used to measure the quantity or amount of something. These data types answer questions like "How much?" or "How long?" and can be useful for mathematical calculations and statistical analyses. Key characteristics of quantitative data include being numerical, measurable, countable, and arithmetical.

- **Numerical:** Quantitative data is expressed in numbers, representing a measurable quantity. Examples include heights, weights and test scores.



- **Measurable:** Quantitative data can be measured using instruments or tools. Examples include using a ruler for length and a thermometer for temperature.
- **Countable:** Quantitative data can be counted or enumerated, particularly for discrete data. Examples include the number of students the number of cars.
- **Arithmetical:** Quantitative data can be used in arithmetic operations. For instance, multiplying the unit price of a fruit by its weight to find the total price, or calculating annual school fees by multiplying monthly fees by the number of months.

6.3 Organising and Analysing Data

Organizing data systematically is very important for clear analysis and interpretation. When data is well-organised, it helps reduce errors. For example, imagine you have a list of students and their test scores, a messy, list might lead to accidentally recording a score under the wrong student's name. Organizing the data neatly in a table reduces such mistakes.

Importance of Organising Data

- Proper organisation saves time. Think about searching for a book in a messy room versus a neatly arranged bookshelf. Similarly, when data is organised, it's easier to find and analyse.
- Organised data improves clarity. When data is presented clearly, it's easier to understand and interpret.
- By organising data into tables, charts, and graphs, you can quickly grasp what the data is saying, making it simpler to draw conclusions and make decisions.

To make data easy to understand, it can be organised into tables, charts, and graphs. Here's how:

Data Tables: Imagine you have data about students' scores in different subjects. An example of this data is shown in table 6.1.

Student	Math	Science	English
Ali	85	78	90
Sara	78	88	85
Ahmed	92	82	87
Fatima	90	80	89
Bilal	67	75	70

Table 6.1

Tables like this help present data clearly, making it easy to compare and analyse the scores of different students across various subjects.



Charts: Charts are visual representation of data designed to make complex information easier to understand. Charts help identify patterns, trends and outliers in datasets. Common types of charts include:

- Bar charts
- Line charts
- Pie charts

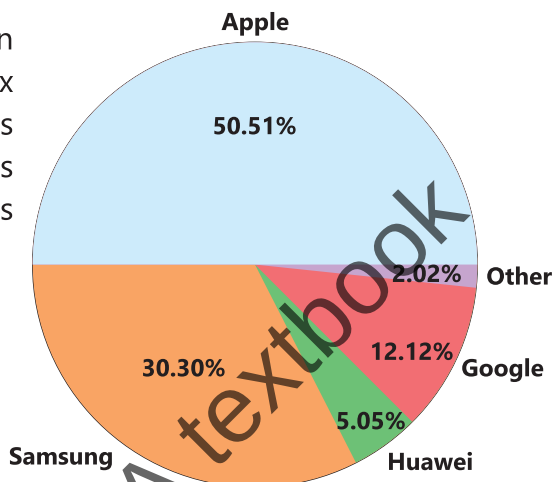


Fig 6.1 chart

Graphs: Graphs are visual tools used to represent data and show relationship between different data points. Common types of graphs include:

- Line graphs
- Bar graphs
- Scatter plots
- Histograms etc



Fig 6.2 Graphs

9.3.1 Methods and Tools of Data Collection

Data collection is the process of gathering information to answer questions, make decisions, or understand something better. There are different methods and tools for collecting data, each with its own way of gathering and recording information.

Methods of Data Collection

Here's a simple explanation of the main methods:

• **Surveys:** Surveys collect information from people by asking them questions. This can be done on paper, over the phone, or online. For example, to find out your classmates' favourite ice cream flavours, you might create a survey with questions like "What is your favourite ice cream flavour?" and give it to your classmates to fill out.



Best Practices in Survey Design and Administration

- Be clear and specific
- Use multiple choice and rating scales
- Test your survey before sending it out
- Keep it short and simple
- Ensure anonymity
- Analyse the results



TidBiTs

- A free tool from Google that helps you create surveys and collect responses online. Link for Google Forms is <https://forms.google.com/>
- Microsoft Office tools allows you to create surveys and quizzes Link for Microsoft Forms is <https://forms.office.com/>
- SurveyMonkey is also a popular tool for creating detailed surveys. Link for SurveyMonkey is <https://www.surveymonkey.com/>

- **Questionnaires:** Questionnaires are similar to surveys but are often written forms that people fill out. They usually have a set of questions that need to be answered. For instance, if your school wants to know which activities students enjoy most, they might give out a questionnaire with questions like, "Which school activity do you enjoy the most? (e.g., sports, art, music)", where students choose their answers from the options provided.
- **Interviews:** Talking to individuals one-on-one to gather detailed information. Example: Interviewing a school teacher to understand their experience and challenges.
- **Observations:** Watching and noting what happens in a particular situation. Example: Observing how students behave during a group project to understand how they work in a team.
- **Online Data Sources:** These include websites, databases, and digital tools where you can find and collect information. For example, if you're researching the most popular pets, you might use the internet to find statistics or articles about pet ownership.

ACTiViTY



Data Collection and Organization

Activity Type: Individual

Activity Task Details:

• Create Survey:

1. Design a short survey with clear, specific questions on a topic of choice (e.g., favourite school subject).



2. Ensure questions are designed to gather meaningful data.

• **Collect Responses:**

1. Share the survey link with classmates or a small group to gather responses.
2. Collect data over a specified period.

• **Organise Data:**

1. Enter the collected data into a spreadsheet.
2. Organise the data into tables and use basic spreadsheet functions to clean and prepare it.

• **Create Visuals:**

1. Use spreadsheet tools to create at least one chart or graph (e.g., bar chart, pie chart).
2. Label charts clearly and ensure they represent the data accurately.

• **Share Findings:**

1. Present the findings to the class.
2. Explain the visuals and what they reveal about the data.

6.4 Data Types

With respect to storage and processing, data has two types: structured and unstructured data.

Structured Data

Structured data is organised and formatted to be easily searchable and analysable. Examples include data in spread sheets and traditional databases as shown in table 6.2.

Student ID	Student Name	Class	Date of Birth	Fee Status	Height
001	Ali Akbar	9th	3/25/2009	Paid	4.7
002	Faheem Aslam	9th	5/7/2008	Paid	4.9
003	Munir Ahmad	9th	6/11/2009	Unpaid	5.2
004	Khalid Mahmood	9th	9/13/2009	Paid	5.6
005	Kamran Malik	9th	7/21/2009	Paid	5.3

Table 6.2 Structured data



Unstructured Data

Unstructured data is more free-form and doesn't fit into a specific format. Examples include text from emails, social media posts, videos, and images.

6.5 Data Visualization

Data visualization is the process of turning numbers and information into pictures. These pictures make it easier for us to understand what the data is telling us. When we look at data in the form of charts or graphs, it becomes simpler to see patterns, trends, and relationships.

Data Visualization Tools

Some popular tools for visualizing data include Microsoft Excel, Google Sheets and Tableau and Microsoft Power BI.

- **Microsoft Excel:** A spreadsheet tool that can create various charts and graphs.
- **Google Sheets:** Similar to Excel, it allows you to create and share visualizations online.
- **Tableau:** A powerful tool specifically designed for creating detailed and interactive visualizations.
- **Microsoft Power BI:** Power BI allows users to create a wide variety of visualizations, including charts, graphs, maps, and more, to represent data in an intuitive and understandable format.

6.6 Data Pre-Processing and Analysis

Data Pre-processing

Data pre-processing is the first and most important step in working with data. It involves getting the data ready for analysis by cleaning and organizing it. Think of it like preparing ingredients before cooking a meal; you need to wash, chop, and measure everything so that the cooking process goes smoothly.

Data Pre-processing Techniques

In data pre-processing, we use various techniques to ensure the data is accurate and ready for analysis. Here are some important steps and techniques:

i) Evaluating Data Quality:

Before using data, we need to check its quality, ensuring it is accurate, complete, and reliable. We ask questions like: Is any data missing? Are there errors or incorrect entries? Is the data consistent and up-to-date?

Example

Imagine you have a list of students' names and their test scores. You need to



check if all students have scores recorded, if the scores are correct, and if they are from the current session.

ii) Identifying Errors, Outliers, and Biases:

Errors are mistakes in the data.

Example

Check, if the data is within a valid range. For instance, if maximum marks of a subject are 100 and a student's score is recorded as 105, it's clearly an error because scores can't exceed 100.

Outliers are unusual or extreme values that don't fit the pattern of the rest of the data.

Example

In a list of test scores, if most students scored between 50 and 80, but one student scored 5, the score of 5 is an outlier.

Biases are distortions that affect the accuracy of the data.

Example

If a survey only includes answers from students in one school, it may not represent the opinions of all students in the city. Hence if you want to analyse opinion of entire city students then this data would be biased due to the limited sample.

Implementing Data Validation and Cleaning Processes

Once we identify issues, we need to fix them by validating and cleaning the data.

Data Validation involves checking data completeness and accuracy.

Validating data completeness means to ensure that no data is missing.

For Example making sure every student has a test score recorded.

Validating data accuracy ensures the data is correct. **For example** verifying that all test scores are between 0 and 100.

Data Cleaning involves error removing, handling missing data and dealing with outliers. Errors are either corrected or the incorrect data is deleted.

For example changing a score of 105 to a valid number like 95, or if the valid number is not known, deleting this record from data.

To handle missing data, decide how to deal with gaps for example if a student's test score is missing, you might use the average score of the class to fill in the gap.

To **deal with Outliers**, decide whether to keep, change, or remove unusual values. **For example** investigating why a student scored 5 and determining if it's an error or a true value.



Data Analysis Techniques

Data analysis involves examining data to discover useful information, draw conclusions, and support decision-making. There are two main types of data analysis: quantitative and qualitative.

Quantitative Analysis

Quantitative analysis deals with numbers and measurable data. It helps us understand patterns, relationships, and trends in numeric data.

Qualitative Analysis

Qualitative analysis deals with non-numeric data such as text, images, and sounds. It helps us understand meanings, concepts, and experiences.

6.7 Cloud Storage and Data Backups

In today's world, working together and sharing information online is essential. Collaborative tools make this possible, allowing people to work together, share files, and access their work from anywhere.

Cloud Storage for Data Management

Using cloud storage for managing data has become an essential part of how we store, access, and share information. It allows us to save files on the internet for access from any device.

Remote Access

- Remote access refers to the ability to connect to and use a computer or network from a distant location. You can work on files, use software, or access resources on a computer or network that is not physically near you.

Did You Know?

Cloud Storage Evolution: Storing files in the cloud is like having an infinite bookshelf. Instead of cluttering your physical space, your documents, photos, and videos can be stored online, freeing up room for more important things.

Data Backups

Data Backups are copies of important data or files stored separately from the original to protect against data loss. Backups are essential to ensure that you can recover your data if something goes wrong, such as accidental deletion, hardware failure, or a computer virus.



Did You Know?

Global Collaboration: Cloud-based tools have enabled people to work together on projects from different parts of the world. For instance, a student in Pakistan can collaborate on a science project with peers in the USA and Australia simultaneously.

For example, if you are working on a school project and save it on your computer. To ensure you don't lose your work, you create a backup by saving a copy on Google Drive or a USB.

- **Automatic Backups**
- **Restore Files**
- **Resume Work**

Benefits of Collaborative Tools

1. **Enhanced Productivity:** When multiple students work on different sections of a project simultaneously the project gets done faster. This saves time and allows everyone to contribute their best work.
2. **Version Control:** Google Docs automatically saves every change, so you can go back to previous versions if needed. You don't lose work, and you can see who made which changes in the document.

6.8 Introduction to Data Science

Data science is like being a detective, but instead of solving crimes, you solve problems using data. Imagine you want to figure out why some students do better in exams than others. Data science helps you gather information about students' study habits, look for patterns, and find out what makes a difference. For example, you might discover that students who study in groups tend to score higher. By using data science, you can make better decisions and find solutions to everyday problems. It combines computer skills, math, and business knowledge to turn raw data into useful insights, just like turning ingredients into a delicious recipe.

Understanding Data Science

Knowing about data science is important because it helps us make better decisions in our daily lives. For example, if you understand how to analyse data, you can figure out the best time to study when you're most focused. Businesses use data science to learn what their customers like and improve their products. Sports teams use data to create better strategies and enhance their performance. Even doctors use data science to predict health trends and provide



better care. By learning data science, you can solve problems more effectively and make smarter choices, whether at school, work, or in everyday life.

Interdisciplinary Nature of Data Science:

Data science is unique because it combines different fields to work with data effectively. It includes:

- **Computer Science:** For handling and organizing data.
- **Mathematics and Statistics:** For analysing data and finding patterns.
- **Business Knowledge:** For applying these insights to real-life problems and making informed decisions.

Tools in Data Science

Data science relies on various tools and technologies to help manage, analyse, and visualize data effectively. These tools and techniques are essential for turning raw data into useful insights.

- **Microsoft Excel:** Excel is a widely-used spreadsheet program that helps you organise and analyse data. For example, if you have a list of your test scores, you can use Excel to create graphs that show your progress over time. This makes it easy to see how you're improving and identify any patterns in your performance.
- **Python:** Python is a popular programming language used in data science for analysing data and building data models. It comes with powerful libraries like Pandas, for data manipulation and Matplotlib for creating graphs. For instance, you might use Pandas to analyse data from a survey and then use Matplotlib to create a chart that shows the results.
- **R:** R is another programming language designed for statistical analysis and data visualization. It is particularly good at handling complex data and presenting it in a clear way. For example, you might use R to analyse a dataset from a scientific experiment and create detailed plots that make the results easier to understand.
- **SQL:** SQL (Structured Query Language) is used to manage and query databases. For instance, if you need to find all students who scored above 90 on a test from a large database, SQL allows you to extract this specific information quickly. It's like asking a librarian to find all books by a particular author from a huge collection.

These tools are important for data scientists as they help in make sense of large amounts of data, provide valuable insights, and support informed decisions.



Summary

- Data refers to raw facts we gather about things around us, which can then process to extract useful information.
- Qualitative data refers to categories or labels that describe qualities or characteristics rather than quantities.
- Quantitative data are numbers used to measure the quantity or amount of something.
- Structured Data is organized and formatted for easy searching and analysis: examples include data in spread-sheets and traditional databases.
- Unstructured Data is more free-form and doesn't fit into a specific format.
- Data visualization turns numbers and information into visual representations, making data easier to understand.
- Data pre-processing is the initial and important step in working with data. It involves cleaning and organizing it for analysis.
- Data analysis involves examining data to discover useful information, draw conclusions, and support decision-making.
- Qualitative analysis deals with non-numeric data such as text, images, and sounds. It helps us understand meanings, concepts, and experiences.
- Remote access refers to the ability to connect to and use a computer or network from a distant location.
- A backup is a copy of important data or files stored separately from the original, used to protect against data loss.
- Data science is like being a detective, solving problems using data instead of solving crimes.
- Data science involves gathering, analysing, and interpreting large amounts of data to find patterns and useful information.
- Data Analytics is a branch of data science focussed on analysing data to understand it better and support decisions making.



EXERCISE

Multiple Choice Questions (MCQs)

1. What is data?

- a) Processed information
- b) Raw facts gathered about things
- c) A collection of numbers only
- d) A list of observed events

2. Which of the following is an example of qualitative data?

- a) Temperature readings in degrees Celsius
- b) Number of students in a class
- c) Favourite ice cream flavours
- d) Test scores out of 100

3. How can you organise data to make it easier to analyse?

- a) By writing it in long paragraphs
- b) By creating tables, charts, and graphs
- c) By storing it in random files
- d) By keeping it in a messy notebook

4. What is the main purpose of data collection?

- a) To create random numbers
- b) To gather information to answer questions or make decisions
- c) To delete old data
- d) To format text documents

5. What is the primary purpose of data visualization?

- a) To generate random numbers
- b) To convert text into data
- c) To make data easier to understand by turning it into pictures
- d) To hide complex data

6. Which tool is specifically designed for creating detailed and interactive visualizations?

- a) Microsoft Excel
- b) Google Sheets
- c) Tableau
- d) PowerPoint



7. What is the first step in the data science process?

- a) Data Cleaning
- b) Data Analysis
- c) Data Collection
- d) Understanding the problem

Short Questions

1. Write two differences between qualitative and quantitative data?
2. Which method would you use to collect opinions from a large group of people about a new school policy?
3. What type of data is the number of students in your class?
4. Why is it important to organise data into tables or charts before analyzing it?
5. Explain why data visualization is important.
6. For which purpose do we use line graph?
7. How does Microsoft Power BI help in data visualization?
8. How can we use Python in Data Science?
9. What is SQL?

Long Questions

1. Explain the differences between qualitative and quantitative data.
2. Describe the process of conducting a survey to gather data about students' favourite extracurricular activities.
3. Analyse the benefits and challenges of using digital tools like Google Forms for data collection.
4. Explain the role of tables and charts in data analysis. Provide examples.
5. Discuss the importance and benefits of data visualization.
6. Why is it essential for businesses and decision-makers to use charts, graphs, and dashboards?



UNIT 7

Emerging Technologies in Computer Science



Student Learning Outcomes

By the end of this chapter, you will be able to:

- Define Artificial Intelligence (AI).
- Identify various applications of AI in different domains such as healthcare, education, and gaming.
- Explain the subfields of AI, including machine learning, natural language processing, computer vision, and robotics.
- Understand the role of AI techniques in advancing machine learning models and other capabilities.
- Define the Internet of Things (IoT) and understand its significance in connecting devices and systems.
- Describe the components of IoT systems, including sensors, devices, networks, and data analysis.
- Explore various applications of IoT in domains such as smart homes, healthcare, and transportation.
- Discuss security and privacy considerations in IoT deployments.
- Analyze the potential risks and challenges associated with AI and IoT, including data privacy and algorithmic bias.
- Examine policy and regulatory frameworks addressing the challenges of AI and IoT. Evaluate the social impact of AI and IoT on daily life, work environments, and society at large.
- Develop strategies for addressing ethical concerns and maximizing the social benefits of AI and IoT.



Introduction

Artificial Intelligence (AI) is a rapidly growing field that is changing various aspects of our lives. These technologies are not only reshaping industries but also redefining the way we live, work, and interact with our environment. This chapter provides an introduction to AI and IoT, explores their applications, discusses the algorithms and techniques that power them, and examines the broader implications and future prospects of these transformative technologies.

7.1 Introduction to Artificial Intelligence (AI)

Artificial Intelligence is a rapidly growing field that is transforming various aspects of our lives. From healthcare to gaming, AI technologies are being applied to solve complex problems and improve our daily experiences.

Artificial Intelligence denotes the simulation of human thinking ability in computer systems to think and learn in a manner like humans.

Did You Know?

The first AI program called the Logic Theorist, was created in 1955 by Allen Newell and Herbert A. Simon. It was designed to mimic the problem-solving skills of a human being.

Applications and Subfields

Applications of AI in Various Domains:

AI has numerous applications across different fields:

- **Healthcare:** AI is used for diagnosing diseases, personalizing treatment plans, and predicting patient outcomes.
- **Education:** AI-powered tools provide personalized learning experiences, automate administrative tasks, and offer insights into student performance.
- **Gaming:** AI enhances game design, creates realistic characters, and improves player experiences.
- **Transportation:** Self-driving cars and traffic management systems rely on AI to improve safety and efficiency.
- **Automobile:** AI is transforming the automotive industry by enabling autonomous driving, advanced driver assistance, and optimization of vehicle performance and maintenance. The integration of AI-powered systems is



improving safety, efficiency, and user experience in modern vehicles.

- **Finance:** AI is transforming the finance industry by enabling personalized investment recommendations, fraud detection, algorithmic trading, process automation, and risk assessment to enhance decision-making, improve efficiency, and provide customized financial services.
- **Social media:** AI is extensively used in social media to power personalized content recommendations, automated content generation, sentiment analysis, user behavior prediction, and targeted advertising to enhance user engagement and optimize marketing strategies.
- **Agriculture:** AI is transforming agriculture by enabling precision farming techniques, such as predictive analytics for crop yields, automated irrigation systems, and computer vision for disease and pest detection.
- **E-Commerce:** AI is highly integrated into e-commerce platforms, powering personalized product recommendations, intelligent chatbots for customer support, fraud detection systems, and others.

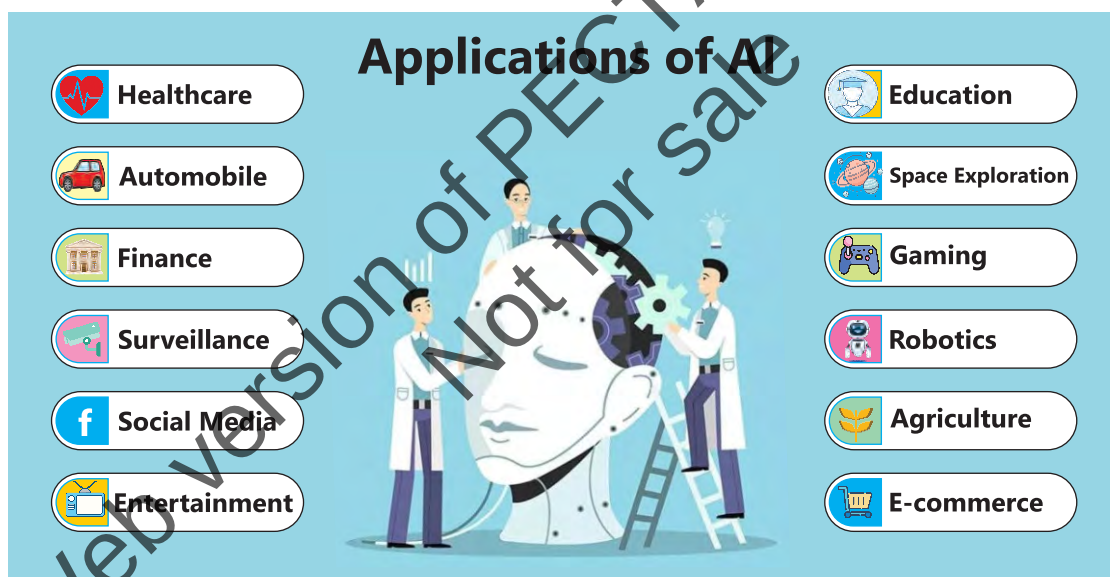


Figure 7.1: Applications of AI in Different Domains

• Subfields of AI:

AI encompasses several subfields, each focusing on different aspects of intelligence and technology:

- **Machine Learning:** Machine learning is a type of artificial intelligence where computers learn from experience and improve over time without being explicitly programmed. It's like teaching a computer by showing it lots



of examples, and it figures out how to do things on its own.

- **Deep Learning:** Deep learning is a special kind of machine learning. It uses complex structures called neural networks, which are inspired by how our brains work. These networks help computers learn from lots of data and make decisions or recognize patterns even better.
- **Natural Language Processing (NLP):** Natural Language Processing, or NLP, is a technology that helps computers understand and talk to us in our language. It's like teaching a computer to read, write, and even chat with us.

Example: When you ask Siri or Alexa a question, they use NLP to understand what you're saying and give you a helpful answer. Another example is when you type a message and your phone suggests words to complete your sentence. That's NLP at work!

- **Computer Vision:** Computer vision is a field of artificial intelligence that enables computers to see and understand the visual world. It helps computers interpret images and videos.
- **Robotics:** Robotics is the science of building and programming robots. Robots are machines that can do tasks for us, like cleaning the floor or building cars. Some robots can even think and make decisions.

ACTiViTy



Research and present a real-world application of AI in one of the domains already mentioned. Explain how AI is used and what benefits it brings.

7.2 AI Algorithms and Techniques

Artificial Intelligence (AI) involves using algorithms and techniques to enable machines to perform tasks that typically require human intelligence. In this section, we will explore different types of AI algorithms and understand their roles in advancing the capabilities of AI, particularly through machine learning models.

Did You Know?

In Pakistan, robots and machines are being used to perform medical operations?

Types of AI Algorithms:

AI algorithms can be broadly categorized into two types based on their



interpretability: (I) explainable (whitebox) (ii) unexplainable (blackbox) algorithms.

Explainable (Whitebox) Algorithms

Explainable or whitebox algorithms are those where the decision-making process is transparent and understandable. These algorithms allow users to see and understand how decisions are made.

Unexplainable (Blackbox) Algorithms

Unexplainable or blackbox algorithms are those where the decision-making process is not easily interpretable. These algorithms often involve complex computations and interactions that make it difficult to understand how a particular decision was reached. Examples include neural networks and deep learning models, which we have already discussed in previous Section.

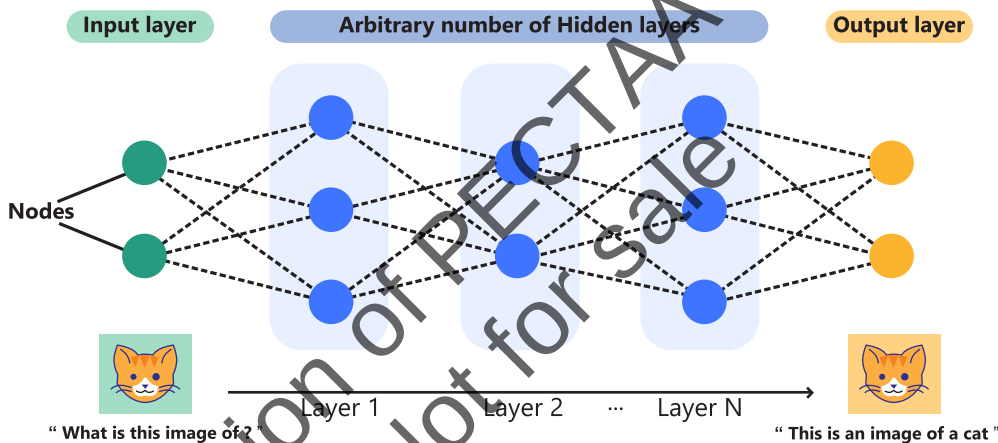


Figure 7.2: Example of a Neural Network

Did You Know?

Google's AlphaGo, a reinforcement learning model, made headlines by defeating a world champion in the game of Go, a feat considered nearly impossible due to the game's complexity.

7.3 Introduction to Internet of Things (IoT)

Understanding IoT

IoT is a revolutionary concept that is transforming the way we live and work. It involves connecting everyday devices and systems to the internet, allowing them to communicate and interact with each other.

Definition and Components

Definition:

IoT is a network of physical objects, or "things," that are equipped with sensors, software, and other technologies to facilitate the exchange of data with other



devices and systems over the internet. This facilitates the development of new, innovative services and the implementation of more intelligent, efficient operations.

Significance of IoT:

IoT is significant because it allows for the seamless integration of the physical and digital worlds. This connection enables devices to collect and share data, which can be analysed to improve efficiency, provide better services, and create new opportunities in various fields such as healthcare, agriculture, and smart homes.

Components of IoT Systems:

An IoT system typically consists of the following components:

- **Sensors:** These are devices that detect and measure physical properties like temperature, humidity, light, and motion. Sensors collect data from the environment.
- **Actuators:** These are devices that convert energy into motion. In IoT, an actuator can act on data to generate output.
- **Devices:** These include everyday objects like smartwatches, refrigerators, and cars that are connected to the internet. Devices use the data collected by sensors to perform specific tasks.
- **Networks:** These are the communication pathways that connect sensors and devices to the internet, allowing them to share data. Networks can be wired or wireless.
- **Data Analysis:** This involves processing and analysing the data collected by sensors to gain insights and make decisions. Data analysis can be done on the device itself, in the cloud, or on a central server.

ACTIVITY



Prepare a list of devices that convert energy into motion. In IoT, an actuator can act on data to generate output.

Did You Know?

Did you know? The term "Internet of Things" was coined by Kevin Ashton in 1999 during his work at Procter & Gamble.



TidBiTs

Always ensure that your IoT devices are connected to a secure network to protect your data from unauthorized access.



Example: Smart Home System

One of the practical uses of IoT is a smart home system. There are several internet-connected appliances in a "smart home," including the temperature control system, lighting, and surveillance cameras.

ACTIVITY



Think about your own home. What everyday devices could be connected to the internet to make your home smarter? Discuss with your classmates and draw a simple diagram showing how these devices could be connected in an IoT system.

Did You Know?

Did you know? Smart home devices can help save energy by automatically turning off lights and adjusting heating or cooling when you're not home.

IoT Applications

The Internet of Things (IoT) is transforming many aspects of our lives by connecting devices and systems in various domains. Let's explore some of the exciting applications of IoT and understand the importance of security and privacy in these deployments.

Healthcare:

IoT is revolutionising healthcare by providing better patient monitoring and care. IoT devices can track vital signs, remind patients to take medication, and alert healthcare providers in case of emergencies.

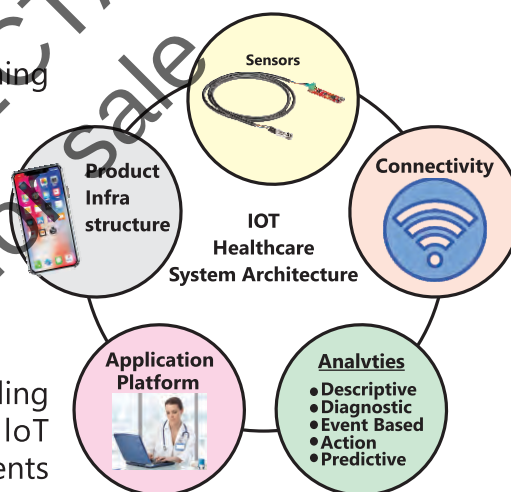


Figure 7.3: Example of IoT in Healthcare

Always use devices from reputable manufacturers and keep your health data secure using strong passwords and updating your devices regularly.

Tidbits

Transportation:

IoT is enhancing transportation systems, making them more efficient and safer. Connected vehicles, smart traffic lights, and real-time tracking systems are some examples of how IoT is used in transportation.

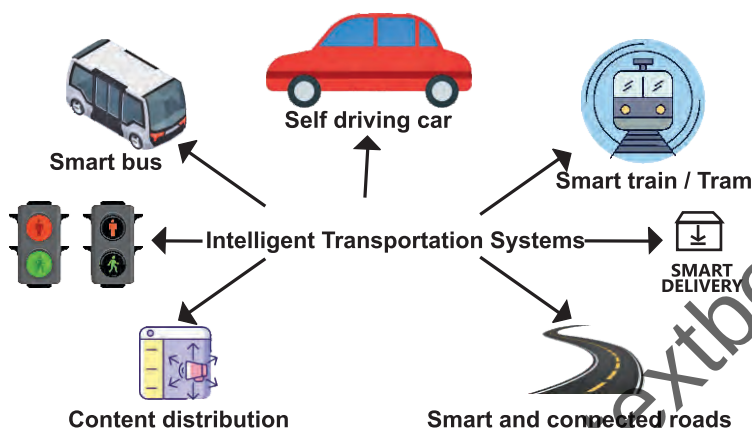


Figure 7.4: Example of IoT in Transportation

ACTiViTY



Think about how IoT can improve your school. Can you come up with ideas for smart classrooms, smart libraries, or even smart buses? Discuss with your classmates and draw a diagram showing your IoT-enabled school.

Security and Privacy Considerations in IoT Deployments:

While IoT offers many benefits, it also raises security and privacy concerns. As more devices are connected to the internet, the risk of cyber-attacks increases. It's important to ensure that IoT systems are secure to protect personal data and privacy.

Examples of Security Measures:

- **Strong Passwords:** Use strong, unique passwords for all IoT devices to prevent unauthorized access.
- **Regular Updates:** Keep the software and firmware of your IoT devices up to date to protect against known vulnerabilities.
- **Encryption:** Ensure that data transmitted between devices is encrypted to protect it from being intercepted by hackers.

Did You Know?

In 2020, there were over 20 billion IoT devices in use worldwide, highlighting the rapid growth and importance of IoT technology.



Summary

- Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think and learn like humans.
- Machine learning is a type of artificial intelligence where computers learn from experience and improve over time without being explicitly programmed.
- Deep learning is a special kind of machine learning. It uses complex structures called neural networks, which are inspired by how our brains work.
- Natural Language Processing, or NLP, is a technology that helps computers understand and talk to us in our own language.
- Computer vision is a field of artificial intelligence that enables computers to see and understand the visual world.
- Robotics is the science of building and programming robots. Robots are machines that can do tasks for us, like cleaning the floor or building cars.
- AI algorithms can be broadly categorized into two types based on their interpretability: explainable (whitebox) and unexplainable (blackbox) algorithms.
- Whitebox algorithms are those where the decision-making process is transparent and understandable.
- Blackbox algorithms are those where the decision-making process is not easily interpretable.
- The Internet of Things (IoT) refers to the network of physical objects-“things”-that are embedded with sensors, software, and other technologies to connect and exchange data with other devices and systems over the internet.



EXERCISE

Multiple Choice Questions

1. Which of the following is not a subfield of AI?

- a) Machine Learning
- b) Natural Language Processing
- c) Computer Vision
- d) DBMS

2. Which of these is a security concern in IoT deployments?

- a) Device vulnerability
- b) Data privacy
- c) Lack of standardization
- d) All of the above

3. Which of the following is an application of AI in healthcare?

- a) Personalized drug development
- b) Automated diagnosis
- c) Remote patient monitoring
- d) All of the above

4. What is the primary purpose of using AI techniques in machine learning models?

- a) To improve accuracy
- b) To enhance interpretability
- c) To reduce computational complexity
- d) All of the above

5. What is the key difference between explainable (whitebox) and unexplainable (blackbox) AI models?

- a) The complexity of the model
- b) The ability to understand the decision-making process
- c) The performance of the model
- d) The training data used

6. Which of the following is an application of IoT in the transportation domain?

- a) Smart traffic management
- b) Vehicle-to-Vehicle (V2V) communication
- c) Predictive maintenance of vehicles
- d) All of the above



Short Questions:

1. Define Artificial Intelligence (AI).
2. Provide two examples of AI applications in healthcare.
3. Explain the role of AI techniques in advancing machine learning models.
4. Define the Internet of Things (IoT).
5. Describe the significance of IoT in connecting devices and systems.
6. Describe applications of IoT in Transportatin Domain.

Long Questions:

1. Discuss the various applications of AI in the field of education.
2. Describe the components of an IoT system.
3. Write a note on sub-fields of AI.

Web version of PECTAA textbook
Not for sale



UNIT

8

Ethical, Social, and Legal Concerns in Computer Usage



Student Learning Outcomes

By the end of this chapter, you will be able to:

- Describe the importance of using computers safely and responsibly in daily activities.
- Explain how to protect personal information while using computers and online platforms.
- Identify the factors to consider when selecting computer hardware and software for safety, efficiency, and compatibility.
- Demonstrate how to create strong and unique passwords.
- Identify concept and benefits of Two-Factor Authentication (2FA), risks of using public Wi-Fi for sensitive transactions.
- Identify and avoid common online scams and phishing attempts.
- Understand importance of regularly backing up important data.
- Describe responsible behavior on social media, email, cloud services, and online applications.
- Explain the significance of privacy settings and data security measures in protecting personal information online.
- Identify ethical and legal responsibilities related to intellectual property rights, including copyright, trademarks, and patents.
- Describe ethical guidelines for data collection, storage, and sharing, including the importance of informed consent and data security.



Introduction

Computers and digital devices are now a big part of our daily lives, helping us with schoolwork, staying in touch with friends, and even playing games. But with all these great tools comes the need to use them safely and responsibly. In this chapter, we'll learn how to use computers and digital platforms wisely. We'll explore how to keep our personal information safe, make smart choices about the hardware and software we use, and behave respectfully online. By understanding these important aspects, we can enjoy technology while avoiding problems and protecting ourselves and others. Let's understand the world of secure upright computer usage to make sure we get the most out of our digital experiences!

8.1 Secure Computer Usage

Computers have become a crucial part of our everyday lives. Whether we are using them for schoolwork, chatting with friends, or playing games, it's important to use computers safely and responsibly. Safe and responsible computer usage means knowing how to protect our personal information, making wise choices about the hardware and software we use, and ensuring that our online behavior is respectful and ethical. In this chapter, we will explore the key aspects of using computers wisely and securely, so we can enjoy the benefits of technology while avoiding potential risks.



TidBiTs

Online Etiquette: The "Golden Rule" applies online too, treat others as you want to be treated. This helps maintain respectful and positive interactions.

8.2 Responsible Computer Usage

Being responsible with technology means making thoughtful decisions when using computers. This includes selecting the right hardware and software, safeguarding our data, and using the internet in a way that respects others. Let's take a closer look at why it's important to use technology responsibly.

When we talk about computers, hardware refers to the physical parts like the monitor, keyboard, and CPU, while software includes the programs and applications we use, such as word processors or games. Choosing the right hardware and software is important because it can affect the safety, efficiency, and overall experience of using a computer.

- **Safety:** Just as you wouldn't use a broken or unsafe tool, using outdated or insecure hardware or software can put you at risk. For example, using a



computer without updated antivirus software can make it easier for viruses or hackers to steal your information.

- **Efficiency:** The right hardware and software help you complete tasks quickly and easily. Imagine trying to play a new video game on an old computer, the game might be slow or not work at all, making it frustrating to use.
- **Compatibility:** This means that the hardware and software should work well together. You should always check the system requirements on software packages and match them with your computer's specifications to ensure compatibility.

Use of Secure Digital Platforms

Using digital platforms securely means taking extra steps to protect your information and ensuring that your online activities do not put you or others at risk. Here are some tips:

Using Strong Passwords: Always create strong, unique passwords for your accounts. A strong password typically includes a mix of letters, numbers, and special characters. For example, instead of using "password123," you could use something like "B3tt3rP@ssw0rd!".

Two-Factor Authentication (2FA): Two-Factor Authentication adds an extra layer of security to your accounts. After entering your password, you will be asked to provide another piece of information, like a code sent to your phone. This makes it much harder for someone to hack into your account.

Avoiding Public Wi-Fi for Sensitive Transactions: Public Wi-Fi networks, like those in cafes or libraries, are often less secure. It's best to avoid accessing sensitive information, such as online banking, while connected to these networks. Instead, wait until you're on a secure, private network at home.

Being Aware of Scams: Online scams are designed to trick you into giving away your personal information. These can include phishing emails that pretend to be from legitimate companies asking for your login details. Always be skeptical of unsolicited requests for personal information, and verify the source before responding.

Regularly Reviewing Your Account Activity: Periodically check your online accounts for any unusual activity. This includes looking at your recent logins, messages, and transactions. If you notice anything suspicious, such as logins from unfamiliar locations, change your password immediately and report the activity to the platform.

Backing Up Important Data: Regularly backing up your data ensures that you won't lose important information if something goes wrong with



your device. You can back up your data to an external hard drive or a cloud storage service like Google Drive or Dropbox.

8.3 Best Practices in Online Behavior

In today's digital world, we use social media, email, cloud services, and online applications every day. While these tools make our lives easier and more connected, it's important to use them responsibly.

Using digital platforms like social media, email, cloud services, and online applications has become a part of our daily lives. However, it's important to use these tools responsibly to ensure our safety and the safety of others.

- **Social Media:** Social media platforms like Facebook, Instagram, and Twitter allow us to connect with friends and share information. But it's important to think before you post. Always avoid sharing personal information, like your home address or phone number, publicly.
- **Email:** Email is a useful tool for communication, especially for school and work. However, it's important to be careful when opening emails from unknown senders. These could contain harmful links or attachments.
- **Cloud Services:** Cloud services like Google Drive or Dropbox allow you to store and share files online. While these services are convenient, it's important to use them wisely. Always use strong passwords to protect your accounts and avoid sharing sensitive information, like passwords or financial details, through cloud storage.
- **Online Applications:** Online applications, such as games, learning apps, or shopping platforms, are fun and useful but can also pose risks. Make sure to download apps only from trusted sources like Google Play Store or Apple App Store to avoid downloading harmful software.

8.4 Intellectual Property Rights

Intellectual property rights are important because they protect the creations and ideas of individuals and organizations. When someone creates something new, like a piece of music, a book, or an invention, they have the right to control how it is used.

Copyright is a legal right that gives creators control over their original works, such as music, books, movies, and software. For example, when an author writes a book, they have the copyright to decide how the book is published, shared, or adapted. This means no one else can copy or distribute the book without the author's permission.

Trademarks are symbols, names, or slogans used by companies to distinguish their products or services from others. For instance, the Nike "swoosh" logo is a



trademark. Trademarks protect brand identity, so no other company can use a similar symbol to confuse customers.

Patents protect new inventions or processes, giving the inventor exclusive rights to make, use, or sell the invention for a certain period. For example, if someone invents a new type of smartphone, they can patent it to prevent others from making or selling a similar phone without permission.

Ethical and Legal Responsibilities Regarding Intellectual Property Rights

Respecting intellectual property rights means understanding that copying, sharing, or using someone else's work without permission is not only unethical but also illegal. For example, downloading movies or software without paying for them is a violation of copyright law. It's important to always seek permission or purchase content legally to support creators and respect their rights.

Legal Compliance in Computing

Software piracy is the illegal copying, distribution, or use of software. When you buy software, you are actually buying a license to use it, not the software itself. Copying it and sharing it with others without a proper license is against the law. Piracy is harmful because it cheats software developers out of the money they need to continue creating and improving their products.

By understanding and respecting intellectual property rights, we can all contribute to a fair and legal digital environment.

Social Networking Safety and Online Interactions

Privacy Settings, Responsible Sharing, and Online Etiquette

Social networking platforms allow us to connect with friends and share our lives, but it's important to do so safely and respectfully.

- **Adjust Your Privacy Settings:** Make sure your social media accounts are set to private, so only people you trust can see your posts. For example, on Instagram, you can set your account to "private" so only approved followers can view your content.
- **Practice Good Online Etiquette:** Be respectful in your online interactions. This means using polite language, avoiding arguments, and not spreading rumors or false information. For example, if you disagree with someone's post, express your opinion respectfully without insulting them.

Addressing Cyberbullying, Harassment, and Respectful Online Interactions

Cyberbullying and online harassment are serious issues that can hurt people emotionally and mentally. Here's how to deal with them:

- **Recognize Cyberbullying:** Cyberbullying involves using the Internet to harm or harass others. This can include sending mean messages, spreading rumors, or posting embarrassing photos of someone without their permission.
- **Report and Block:** If you experience or witness cyberbullying, report it to



the platform and block the person responsible. Most social media platforms have tools to help you do this. For instance, on Facebook, you can block someone to stop them from contacting you or viewing your profile.

- **Support Others:** If you see someone being bullied online, offer your support. You can stand up for them by reporting the bullying or simply offering kind words. This can make a big difference in how they feel.
- **Practice Respectful Interactions:** Always treat others with respect online, just as you would in person. Avoid making negative comments, and think about how your words might affect someone else. For example, if you see a post you don't like, it's better to scroll past it than to leave a hurtful comment.

By using following these guideline for responsible use, you can stay safe, healthy in online world and can enjoy benefits.

Summary

- Safe and responsible computer usage means knowing how to protect our personal information, making wise choices about the hardware and software we use, and ensuring that our online behavior is respectful and ethical.
- Responsible computer usage means protecting ourselves and others when using computers.
- Safe operation of digital platforms and devices, mean using them in a way that protects you from harm and avoids any unwanted issues.
- Using digital platforms securely means taking extra steps to protect your information and ensuring that your online activities do not put you or others at risk.
- Privacy settings and data security measures are essential tools that help protect your personal information while using digital platforms..
- Intellectual property rights are important because they protect the creations and ideas of individuals and organizations.
- Copyright is a legal right that gives creators control over their original works, such as music, books, movies, and software.
- Trademarks are symbols, names, or slogans used by companies to distinguish their products or services from others.
- Patents protect new inventions or processes, giving the inventor exclusive rights to make, use, or sell the invention for a certain period.



EXERCISE

Multiple Choice Questions (MCQs)

1. Why is it important to use computers safely and responsibly?

- a) To ensure we can use them more frequently
- b) To protect our personal information and make wise choices about hardware and software
- c) To make the computer run faster
- d) To avoid paying for software

2. What does "responsible computer usage" include?

- a) Choosing the most expensive hardware
- b) Sharing your passwords with friends
- c) Being cautious about what you share online and protecting yourself and others
- d) Ignoring software updates

3. What should you check to ensure hardware and software compatibility?

- a) The color of the hardware
- b) The system requirements on software packages and match them with your computer's specifications
- c) The price of the hardware
- d) The brand of the hardware

4. Why is it important to use strong, unique passwords?

- a) To make your accounts easy to hack
- b) To help others guess your password
- c) To make it harder for someone to guess your password and access your accounts
- d) To avoid installing software updates

5. What is one reason to avoid clicking on unknown links or downloading files from untrusted sources?

- a) They might contain helpful software
- b) They might contain malware that can harm your device or steal your information
- c) They are usually cheaper
- d) They help your device run faster



6. What does Two-Factor Authentication (2FA) do?

- a) It makes your password easier to guess
- b) It adds an extra layer of security by requiring a second form of verification
- c) It removes the need for a password
- d) It reduces the need for software updates

7. Why should you be cautious when using public Wi-Fi for sensitive transactions?

- a) Public Wi-Fi is usually faster
- b) Public Wi-Fi networks are often less secure
- c) Public Wi-Fi is free
- d) Public Wi-Fi always provides encryption

8. What is an important aspect of responsible use of social media?

- a) Sharing personal information like your home address
- b) Posting photos without considering privacy settings
- c) Being respectful and avoiding sharing sensitive information publicly
- d) Ignoring privacy settings

9. What should you do if you receive an email from an unknown sender asking for personal information?

- a) Provide the information requested
- b) Forward the email to your friends
- c) Ignore or delete the email
- d) Open the email and click on any links

Short Questions

1. What is the importance of using computers safely and responsibly?
2. Why should you create strong, unique passwords for your accounts?
3. How can you protect yourself from harmful links and downloads?
4. What is Two-Factor Authentication (2FA), and why is it useful?
5. Why is it a good idea to avoid using public Wi-Fi for sensitive transactions?
6. Why is it important to be careful about what you share on social media?
7. What is the purpose of privacy laws regarding your personal information?
8. What is software piracy, and why is it harmful?



Long Questions

1. Discuss the importance of responsible computer usage in today's digital world.
2. Describe the steps you should take to ensure the safe operation of digital platforms and devices.
3. Explain the concept of data ethics and its importance in handling personal and sensitive information.

Web version of PECTAA textbook
Not for sale



UNIT 9

Entrepreneurship in Digital Age



Student Learning Outcomes

By the end of this chapter, you will be able to:

- Define entrepreneurship and understand its significance in the digital age.
- Differentiate between various types of entrepreneurs, including startup founders, social entrepreneurs, and intrapreneurs.
- Identify key characteristics of successful entrepreneurs, such as innovation and risk-taking.
- Analyze the impact of digital technologies on entrepreneurship, including digital marketing and e-commerce strategies.
- Apply data analysis techniques to make informed business decisions.
- Generate and develop innovative business ideas through problem identification and creative problem-solving techniques.
- Create comprehensive business plans that include market analysis, revenue models, and digital marketing strategies.
- Understand the principles of ethical and sustainable entrepreneurship and their importance in business operations.



Introduction

Entrepreneurship is the process of designing, launching, and running a new business, often initially a small business, offering a product, process, or service for sale or hire. It involves the willingness to take risks and innovate in order to make business successful. This chapter explores various aspects of entrepreneurship, including its definition, significance, types of entrepreneurs, and the impact of digital technologies. We will also delve into the key characteristics of successful entrepreneurs, the importance of ethical and sustainable practices, and the tools and strategies necessary for navigating the digital landscape.

9.1 Entrepreneurship

Entrepreneurship is the process of starting a new business or organization. It involves identifying a need in the market, coming up with an idea to meet that need, and taking the risk to bring that idea to life. Entrepreneurs are people who create and run these businesses. They are innovators, risk-takers, and problem-solvers.

Did You Know?

Pakistan is home to one of the largest freelancing communities in the world. The country ranks among the top five for freelance growth, with many entrepreneurs offering their skills in areas like software development, graphic design, and digital marketing. This entrepreneurial spirit is helping to drive economic growth and innovation in Pakistan!

Examples of Entrepreneurship

Tech Startups

Tech startups like Facebook, Google, and Apple began as small companies founded by entrepreneurs who had innovative ideas for new technology. These companies have grown to become some of the largest and most influential in the world.



Figure 9.1: Tech startups like Facebook and Apple were founded by entrepreneurs with innovative ideas.



Local Businesses

Local businesses, such as your neighborhood bakery or a small clothing boutique, are also examples of entrepreneurship. These businesses provide products or services to their communities and are often started by individuals with a passion for what they do.



Figure 9.2: Local businesses like bakeries and boutiques are examples of entrepreneurship in your community.

Key Characteristics of Entrepreneurs

Key characteristics refer to the essential traits or qualities that define a particular role or person. In the context of entrepreneurs, key characteristics are those fundamental attributes that are commonly found in successful entrepreneurs. These include:

Did You Know?

The word "entrepreneur" comes from a French word that means "to undertake." Entrepreneurs undertake the task of starting and running new businesses.

Innovation

Innovation means creating something new or improving something that already exists. Entrepreneurs are always looking for new ways to solve problems or make things better.

Risk-Taking

Starting a new business involves risk. Entrepreneurs must be willing to take chances, knowing that they might fail. However, they also know that taking risks can lead to great rewards.

Class ACTiViTy

Think of a problem you face in your daily life. In groups, come up with a business idea that could solve that problem. Present your idea to the class!

Did You Know?

Did you know that Walt Disney, the creator of Mickey Mouse, started his company in his uncle's garage? Today, Disney is one of the largest entertainment companies in the world!



Why is Entrepreneurship Important?

Entrepreneurship is important because it drives economic growth, creates jobs, and fosters innovation. New businesses bring fresh ideas and competition, which can lead to better products and services for everyone.

Economic Growth

When entrepreneurs start new businesses, they contribute to the economy by creating new jobs and providing new products and services. Economic growth means that a country is producing more goods and services, which makes people's lives better. Entrepreneurship plays a big role in this.

Innovation and Progress

Entrepreneurs often come up with groundbreaking ideas that change the way we live and work. For example, the invention of the smartphone has revolutionized communication and access to information.



Figure 9.3: The invention of the smartphone is an example of entrepreneurial innovation that has transformed our lives.

9.2 Entrepreneurship in the Digital Landscape

The digital landscape has revolutionized the way entrepreneurs start and grow their businesses.

Digital Transformation and Entrepreneurship

In the modern world, digital technologies have transformed the way businesses operate. Digital entrepreneurship involves leveraging the power of the Internet, digital platforms, and various technologies to create and run businesses. This section explores the role of digital technologies, digital marketing, and e-commerce in entrepreneurship.

Role of Digital Technologies

Digital technologies provide entrepreneurs with tools and platforms that can enhance their business operations, reach a global audience, and improve efficiency. Examples include social media, mobile apps, cloud computing, and big data analytics.



Social Media

Social media platforms such as Facebook, Instagram, and Twitter allow entrepreneurs to market their products, engage with customers, and build a brand presence.

Class ACTiViTY



Create a social media marketing plan for a hypothetical business. Decide on the target audience, type of content, and platforms you would use. Present your plan to the class!

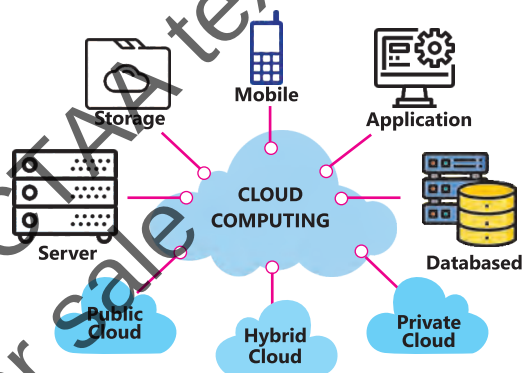


Figure 9.4: Digital technologies like social media and cloud computing are essential tools for modern entrepreneurs.

Mobile Apps

Mobile apps enable businesses to offer services directly to customers' smartphones, providing convenience and enhancing user experience.

Cloud Computing

Cloud computing allows businesses to store data and run applications over the Internet, reducing the need for physical infrastructure and enabling remote work.

Big Data Analytics

Big data analytics helps businesses understand market trends, customer behavior, and make informed decisions based on data insights.

Digital Marketing and E-commerce

Digital marketing and e-commerce have revolutionized the way businesses reach and sell Products and services to customers. Digital marketing involves promoting products and services using digital channels, while e-commerce refers to buying and selling goods online.



Figure 9.5: Mobile apps offer convenience and enhance user experience, making them a valuable tool for digital entrepreneurs.

Digital Marketing Strategies

Digital marketing strategies involve promoting your business online through various channels such as social media, search engines, and email marketing.

- **Search Engine Optimization (SEO):** Optimizing website content to rank higher in search engine results, increasing visibility and traffic.
- **Social Media Marketing:** Using social media platforms to promote products, engage with customers, and build brand awareness.
- **Content Marketing:** Creating valuable content to attract and retain customers, such as blogs, videos, and infographics.
- **Email Marketing:** Sending targeted emails to potential and existing customers to promote products and build relationships.



Figure 9.6: Digital Marketing



Did You Know?

Digital marketing can reach a global audience at a fraction of the cost of traditional marketing methods. It's also easier to measure the effectiveness of digital marketing campaigns using analytics tools.

E-commerce Platforms

E-commerce platforms enable businesses to sell products and services online. They provide tools for setting up online stores, managing payments, and offering customer support.

Setting up and managing online stores, payment gateways, and customer service:

- **Shopify:** Creates and manages online stores with ease.
- **PayPal:** Integrates payment gateways for secure transactions.
- **Zendesk:** Provides customer service solutions for handling inquiries and support.



Figure 9.7: Shopify Store Setup

Did You Know?

Did you know that Pakistan's e-commerce industry is growing rapidly? According to the Pakistan Telecommunication Authority (PTA), the country's e-commerce market size was estimated to be around \$4 billion in 2021, with significant growth expected in the coming years.



9.3 Business Idea Generation

Generating business ideas is an important step in the entrepreneurial journey. This section explores techniques for ideation and problem-solving, focusing on identifying market needs and applying creative problem-solving strategies.

Problem Identification

Identifying market needs and opportunities is the first step in generating a viable business idea. Understanding what people need or want can help you create products or services that fulfill those needs.

Techniques for Identifying Market Needs:

- **Surveys and Questionnaires:** Collecting feedback directly from potential customers.
- **Market Research:** Analyzing market trends and consumer behavior.
- **Observation:** Watching how people interact with existing products and services.

Example: In Pakistan, observing the popularity of online shopping can lead to ideas for new e-commerce platforms tailored to local preferences.



Figure 9.8: Conducting Market Research

Creative Problem Solving

Creative problem-solving involves using innovative thinking to develop solutions to identified problems. This can be done through various methods, including design thinking and brainstorming.

Applying Design Thinking: Design thinking is a solution-focused approach that involves:

1. **Empathizing:** Understanding the needs of those you're designing for.
2. **Defining:** Clearly stating the problem to solve.



3. Ideating: Generating a range of ideas.

4. Prototyping: Creating simple models of solutions.

5. Testing: Trying out prototypes and gathering feedback.

Example: A group of students might use design thinking to develop an app that helps farmers in rural Pakistan access weather forecasts and market prices.

Class ACTiViTy



Form groups and use the design thinking process to create a new product or service. Present your ideas to the class, focusing on how you identified the problem and your innovative solution.

Did You Know?

Many successful companies, like Apple and Google, use design thinking to develop their products, ensuring they meet user needs effectively.

9.4 Developing Business Plans

Developing a business plan is essential for any entrepreneur. A well-crafted business plan provides a roadmap for success, helping to clarify ideas, identify potential challenges, and outline strategies for growth.



Figure 9.9: Business Plan Structure



Creating Comprehensive Business Plans

A business plan is a detailed document that outlines your business goals and the strategies you will use to achieve them. It serves as a blueprint for your business, helping you to stay organized and focused.

Components of a Business Plan

A comprehensive business plan typically includes several key components:

Market Analysis: Market analysis involves researching your target market to understand the needs and preferences of your potential customers. This includes studying market trends, analyzing competitors, and identifying your target audience.

Did You Know

Market analysis can help businesses predict future trends and adapt their strategies accordingly.

Revenue Models: A revenue model outlines how your business will generate income. This includes pricing strategies, sales forecasts, and potential revenue streams.



Figure 9.10: Planning Revenue Models



Figure 9.11: Creating a Prototype

Prototyping and Iteration

Prototyping involves creating a preliminary model of your product or service to test and refine your business idea. Iteration is the process of making improvements based on feedback.

Building Prototypes: Prototypes can be simple sketches, digital models, or even physical samples. The goal is to visualize and test your ideas early on.

9.5 Ethical and Sustainable Entrepreneurship

Entrepreneurship isn't just about making profits; It is also about conducting business in a way that is ethical and sustainable. This section explores the principles of ethical entrepreneurship and how businesses can align their goals with the Sustainable Development Goals (SDGs).

Ethical Practices and Sustainable Growth

Ethical practices in business ensure that a company operates in a way that is fair, transparent, and beneficial to all stakeholders, including customers, employees, and the community. Sustainable growth focuses on developing a business that meets present needs without compromising the ability of future generations to meet their own needs.



Ethical Entrepreneurship

Ethical entrepreneurship involves incorporating principles of ethics into all aspects of business operations and decision-making. This includes honesty, integrity, fairness, and respect for people and the environment.

Principles of Ethics in Business:

- **Honesty:** Being truthful in all business dealings.
- **Integrity:** Acting consistently with moral values.
- **Fairness:** Ensuring equal opportunities and fair treatment.
- **Respect:** Valuing the rights and dignity of all individuals.

Sustainable Development Goals (SDGs)

The Sustainable Development Goals (SDGs) are a set of 17 global goals established by the United Nations to address various social, economic, and environmental challenges. Businesses can play a crucial role in achieving these goals by aligning their strategies with sustainability.

Aligning Business Goals with SDGs:

- **Social Sustainability:** Ensuring fair labour practices, promoting education, and supporting community development.
- **Environmental Sustainability:** Reducing carbon footprints, using renewable resources, and minimizing waste.
- **Economic Sustainability:** Creating jobs, fostering innovation, and contributing to economic growth.

Example: A Pakistani startup focusing on solar energy solutions contributes to SDG 7 (Affordable and Clean Energy) by providing sustainable energy sources to communities with limited access to electricity.



Summary

- **Entrepreneurship:** The process of starting and running a business.
- **Entrepreneur Characteristics:** Key qualities for entrepreneurial success.
- **Digital Entrepreneurship:** Leveraging technology to start and grow a business.
- **Digital Tools:** Utilizing online platforms for business operations.
- **Data-Driven Entrepreneurship:** Making informed decisions based on data.
- **Idea Generation:** Developing new business concepts.
- **Business Planning:** Creating a roadmap for a new venture.
- **Ethical Entrepreneurship:** Balancing profit with social responsibility.

Web version of PECTRA textbook
Not for sale



EXERCISE

Multiple Choice Questions

1. What is entrepreneurship?

- (a) The process of starting a new business or organization
- (b) The process of buying and selling stocks
- (c) The process of working for a large company
- (d) The process of creating a marketing campaign

2. What is a key characteristic of entrepreneurs?

- (a) Avoiding risks
- (b) Seeking job security
- (c) Innovation
- (d) Following established methods

3. An example of a digital technology used by modern entrepreneurs is:

- (a) Newspaper advertisements
- (b) Television commercials
- (c) Mobile apps
- (d) Door-to-door sales

4. Which of the following is NOT a technique for identifying market needs?

- (a) Surveys and Questionnaires
- (b) Market Research
- (c) Observation
- (d) Brainstorming

5. The first step in the design thinking process is:

- (a) Prototyping
- (b) Testing
- (c) Empathizing
- (d) Defining

6. The primary purpose of creating a business plan is:

- (a) To attract investors
- (b) To outline strategies for growth
- (c) To study market trends
- (d) To implement digital marketing strategies

7. What does market analysis involve?

- (a) Calculating revenue forecasts
- (b) Researching competitors and understanding customer needs
- (c) Developing pricing strategies
- (d) Implementing digital marketing campaigns

8. Which Sustainable Development Goal (SDG) focuses on affordable and clean energy?

- (a) SDG5
- (b) SDG7
- (c) SDG12
- (d) SDG17



Short Questions

1. What is the meaning of the word entrepreneur?
2. How do digital technologies help entrepreneurs reach a global audience?
3. What is an e-commerce platform?
4. Why is it important to identify market needs when generating a business idea?
5. What is the main purpose of using design thinking in creative problem-solving?
6. Write the importance of market analysis in a business plan.
7. How can businesses contribute to environmental sustainability according to the Sustainable Development Goals?

Long Questions

1. Explain why entrepreneurship is important for economic growth and innovation.
2. Explain how market research tools, online marketing tools, and e-commerce platforms can work together to help a business succeed in the digital world.
3. Choose a local business in Pakistan and discuss how it aligns with one or more Sustainable Development Goals (SDGs).
4. Explain digital entrepreneurship, also write the principles of ethics in Business.
5. Write down the components of a Business plane.



ANSWER

Unit 1	
1	b
2	b
3	c
4	b
5	c
6	a
7	a
8	c
9	b
10	a

Unit 2	
1	b
2	a
3	b
4	c
5	c
6	c
7	b
8	c
9	c

Unit 3	
1	b
2	c
3	b
4	b
5	b
6	c
7	b
8	d
9	b

Unit 4	
1	b
2	a
3	a
4	c
5	b
6	c
7	b
8	b
9	b
10	b

Unit 5	
1	d
2	a
3	b
4	b
5	a
6	d
7	b
8	b

Unit 6	
1	b
2	c
3	b
4	b
5	c
6	c
7	a

Unit 7	
1	d
2	d
3	d
4	a
5	b
6	d

Unit 8	
1	b
2	c
3	b
4	c
5	b
6	b
7	b
8	c
9	c

Unit 9	
1	a
2	c
3	c
4	d
5	c
6	b
7	b
8	b



Instructions for Preparation of Exam Paper of Computer Science & Entrepreneurship for Grade-9

ESSENTIAL INSTRUCTIONS FOR PAPER SETTERS

The paper of Computer Science & Entrepreneurship for class 9 will consist of 50 marks. The duration for the Multiple-Choice Questions (MCQs) section will be 15 minutes, while the time allocated for the short and long questions section will be 1 hour and 45 minutes. The paper will be made as per following details:

Part-I: Objective:	Q-1: 10 Multiple-Choice Questions. MCQs will be developed from the entire content of the textbook. • One MCQ will be asked from each Chapters No. 1 and 3-9. • Two MCQs will be asked from Chapter No. 2.	1 × 10 = 10
Part-II: Subjective:	This section will contain three short answer questions. Each short answer question will be asked from the content of the textbook. The detail is as follows:	2 × 4 = 8
	Q-2: Four short answer questions have to be answered out of 6. The detail is as follows: • One short answer question should be asked from Chapter No. 1 • Two short answer questions should be asked from Chapter No. 2 • Three short answer questions be asked from Chapter No. 4	



	Q-3: Four short answer questions have to be answered out of 6. The detail is as follows: • One short answer question should be asked from Chapter No. 3 • Two short answer questions should be asked from Chapter No. 5 • Three short answer questions should be asked from Chapter No. 7	
	Q-4: Four short answer questions have to be answered out of 6. The detail is as follows: • Two short answer questions should be asked from Chapter No. 6 • Three short answer questions should be asked from Chapter No. 8 • One short answer question should be asked from Chapter No. 9	
Part-III: Subjective:	This section will contain three detailed questions and students have to attempt two questions carrying 8 marks each. Each detailed question should be asked from the content of the textbook. The detail is as follows. • Q-5: • One detailed question should be asked from Chapter No. 1 • Q-6: • One detailed question should be asked from Chapter No. 3 • Q-7: • One detailed question should be asked from Chapter No. 6 or Chapter No. 9	2 × 8 = 16



Model Paper

Computer Science and Entrepreneurship Class 9

Time Allowed: 15 minutes

وقت: 15 منٹس

Multiple Choice Questions (MCQ)

درست آپشن کا انتخاب کریں۔

- (i) کسی بھی سسٹم کے بنیادی تصورات میں سے ایک ہے۔۔۔۔۔
- (i) One of the fundamental concepts of any system is:
- (A) size (B) age (C) objective (D) price
- عمر (B) سائز (C) قیمت (D) مقصد
- (ii) The Boolean expression represents the AND operation is:
- (A) A+B (B) A.B (C) A.B (D) A⊕B
- ذیل میں سے کون سی بولین ایکسپریشن AND آپریشن کو ظاہر کرتی ہے؟
- (iii) Which of the following step is used to put the plan into action?
- (A) Verify full system functionality (B) Establish a plan of action (C) Implement the solution (D) Document findings and actions
- پورے سسٹم کے کام کی تصدیق (A) عوام کا منصوبہ بنائیں (C) حل کو نافذ کریں (D) حاصلات اور عوام کا مسودہ بنائیں
- (iv) Efficiency of an algorithms is measured in terms of:
- (A) Time Complexity (B) Space complexity (C) Cost Complexity (D) Both a and b
- الگورتھم کی کارکردگی کی پیمائش کس شکل میں کی جاتی ہے؟
- (v) Which device is used to connect multiple networks and direct data packets between them?
- (A) Switch (B) Hub (C) Router (D) Modem
- کون سی ڈیوائس مختلف نیٹ ورکس کو جوڑنے اور ڈیٹا پیکیٹس کو ان میں بھیجنے کے لیے استعمال ہوتی ہے؟
- (vi) Which of the following tag is used for paragraph?
- (A) (B) <p> (C) (D) <pr>
- پیراگراف کے لیے کون سا ٹیگ استعمال ہوتا ہے؟



(vii) کون سا AI الگورتھم قابل وضاحت ماڈل سمجھا جاتا ہے؟

(vii) Which of these AI algorithms is considered and “Unexplainable” model?

(A) Neutral Network

نیورل نیٹ ورک

(B) Decision Trees

ڈیسیزن ٹری

(C) Random Forest

رینڈم فاریسٹ

(D) Linear Regression

لنیر ریگریشن

(viii) ڈیٹا کیا ہے؟

(viii) What is data?

(A) Processed information

پراسس کی گئی معلومات

(B) Raw facts gathered about things

اشیا کے بابت اکٹھی کی گئی خام اعداد و شمار

(C) A collection of numbers only

صرف اعداد کا مجموعہ

(D) A list of observed events

مشاہدہ کی گئی عوامل کی فہرست

(ix) کمپیوٹر کا محفوظ اور ذمہ دار استعمال کیوں ضروری ہے؟

(ix) Why is it important to use computers safely and responsibly?

(A) To ensure we can use them more frequently

یہ یقینی بنانا ہے کہ ہم انھیں زیادہ مرتبہ استعمال کر سکیں

(B) To protect our personal information and make wise choice about hardware and software

ذاتی معلومات کو محفوظ بنانا اور ہارڈ ویئر اور سافٹ ویئر کے متعلق دانشمندانہ فیصلہ کرنا

(C) To make the computer run faster

کمپیوٹر کو زیادہ تیز چلانا

(D) To avoid paying for software

تاکہ سافٹ ویئر کا معاوضہ نہ دینا پڑے

(x) انٹرپرائزمنر کی نمایاں خوبی ہے۔

(x) The key characteristic of entrepreneurs is:

(A) avoiding risks

خطرات سے بچنا

(B) seeking job security

ملازمت کو محفوظ بنانا

(C) innovation

تجربہ

(D) following established methods

طے شدہ طریقہ پر عمل کرنا



Model Paper

Computer Science and Entrepreneurship Class 9

SUBJECTIVE

Total Marks: 40

Time Allowed: 1 hour and 45 minutes

Q. 2: Write short answers to any four (4) questions.

(2x4=8)

- (i) Write characteristics of Von Neumann Architecture (any one).
وان نیومن آرکیٹیکچر کی کوئی ایک خصوصیت لکھیں۔
- (ii) Create logic diagram of the following function?
درج ذیل فنکشن کی لاجک ڈیاگرام بنائیں۔
$$F(x,y,z) = x \cdot \bar{y} + y \cdot \bar{z} + x \cdot z$$
- (iii) Define Security.
سیکورٹی کی تعریف لکھیں۔
- (iv) Write the concept of abstract?
ایبسٹریکٹ کا تصور لکھیں۔
- (v) What is an algorithm?
الگورتھم کیا ہے؟
- (vi) How does Pseudocode help in problem solving?
مسئلہ کے حل میں سوڈو کوڈ کیسے مدد کرتا ہے؟

Q. 3: Write short answers to any four (4) questions.

(2x4=8)

- (i) Define Switch.
سوئچ کی تعریف کریں۔
- (ii) What is the purpose of <head> tag in HTML?
HTML میں <head> ٹیگ کا کیا مقصد ہے؟
- (iii) Define CSS.
CSS کی تعریف کریں۔
- (iv) Enlist name of two subfields of AI.
AI کے دو ذیلی شعبوں کے نام لکھیں۔
- (v) Differentiate between Explainable and Unexplainable AI models.
قابل وضاحت اور ناقابل وضاحت AI ماڈلز میں فرق لکھیں۔
- (vi) Write the role of sensors in IoT.
IoT میں سینسرز کا کردار لکھیں۔



کوئی سے چار سوالات کے مختصر جوابات تحریر کریں۔

Q. 4: Write short answers to any four (4) questions.

(2x4=8)

(i) کو ایٹیسٹوڈیٹا کی تعریف لکھیں۔

(i) Define qualitative Data.

(ii) کلاؤڈ سٹوریج کیا ہے؟

(ii) What is cloud storage?

(iii) ہمیں منفرد اور مضبوط پاس ورڈ کیوں بنانا چاہیے؟

(iii) Why should we create strong and unique password?

(iv) 2FA کیا ہے؟

(iv) What is 2FA?

(v) آپ دھوکہ دہی والی Email کو کیسے پہچانیں گے؟

(v) How can you verify a Scam E-mail?

(vi) Email مارکیٹنگ کیا ہے؟

(vi) What is E-mail marketing?

کوئی سے دو سوالات کے تفصیلی جوابات تحریر کریں۔

Write detailed answers to any two (2) questions.

(2x8=16)

سوال 5: کمپیوٹر کے وان نیومن آرکیٹیکچر کی وضاحت کریں۔

Q. 5:

Explain Von Neumann Architecture of Computer.

سوال 6: OSI ماڈل کی کوئی سی چار لیئرز کی وضاحت کریں۔

Q. 6:

Describe OSI Model layers in Detail (any four).

سوال 7: انٹرپرائز ورلڈ ویب میں ڈیجیٹل ٹیکنالوجیوں کے کردار پر بحث کریں۔

Q. 7:

Discuss the role of digital Technologies in Entrepreneurship.