Based on National Curriculum of Pakistan 2022-23

MODEL TEXTBOOK OF

COMPUTER SCIENCE

GRADE 9

National Book Foundation

NATIONAL BOOK FOUNDATION
AS
FEDERAL TEXTBOOK BOARD, ISLAMABAD

National Book Foundation

Based on National Curriculum of Pakistan 2022-23

Model Textbook of

Computer Science

Grade 9

National Book Foundation

National Curriculum Council
Ministry of Federal Education and Professional Training

National Book Foundation
as
Federal Textbook Board
Islamabad

© 2024 National Book Foundation as Federal Textbook Board, Islamabad
All rights reserved. This volume may not be reproduced in whole
or in part in any form (abridged, photo copy, electronic etc.)
without prior written permission from National Book Foundation

Model Textbook of **Computer Science**
for Grade 9

Authors
Mr. Mohammad Khalid, Dr. Saleem Iqbal, Dr. Faraz Ahsan, Ms. Lubna Kousar,
Mr. Mohammad Sajjad, Mr. Shah Islam, Dr. Hamid Hassan,

Supervision
Dr. Mariam Chughtai
Director, National Curriculum Council
Ministry of Federal Education and Professional Training, Islamabad

IRC Members
Nadeem Mahmood Shaikh, Fazaia Teacher Training Institute Islamabad, Sana Talha, Federal Government Educational Institutions (FGEI) (C/G) Regional Office Chaklala / Cantt, Muhammad Nabeel, Naval Education Headquarter Islamabad, Ms Lubna Zakia, Federal Government Educational Institutions (FGEI) (C/G) Regional Office Chaklala / Cantt, Ms Asma Haroon, Naval Education Headquarter Islamabad
Sadia Mujtaba, Federal Directorate Of Education, Usman Adeel, Fazia Schools And Collages, Ayaz Abdullah
Khubab Hamza, Fazia Schools and Collages, Sadaf Zehra Kazmi, Federal Directorate of Education

IPCW-1 Members
Mr. Muhammad Hashim, School Edu Department, Bolochistan, Sufain Matloob, Directorate Of Curriculum and Teacher Education Khyber Pakhtunkhwa Abbottabad, Mudassar Manzoor Qureshi, Azad Government Of Jammu & Kashmir Directorate of Curriculum Research & Development, Sadaf Zehra Kazmi, Federal Directorate Of Education, Lubna Zakia, Federal Government Educational Institutions (FGEI) (C/G) Regional Office Chaklala / Cantt, Mr, Jahanzaib Khan, Punjab Curriculm & Textbook Board, Qurban Karim,
Mr. Mushtaq Ahmed, Government Of Sindh, School Education & Literacy Deparment

Desk Officer
Zehra Khushal

Management
National Book Foundation

National Book Foundation

TEST EDITION

First Edition - First Impression: 2024 | Pages: | Quantity:

Price: PKR /-
Code: STE-...., ISBN: 978-969-37-
Printer:

Note: All the pictures, paintings and sketches used in this book are only for educational and promotional purpose in public interest.

for Information about other publications of National Book Foundation,
visit our Web Site: www.nbf.org.pk or Phone: 051-9261125
or E-mail: books@nbf.org.pk
to share feedback or correction, please send us an email to nbftextbooks@gmail.com

PREFACE

This Model Textbook has been developed by NBF according to the National Curriculum of Pakistan 2022- 2023. The aim of this textbook is to enhance learning abilities through inculcation of logical thinking in learners, and to develop higher order thinking processes by systematically building upon the foundation of learning from the previous grades. A key emphasis of the present textbook is on creating real life linkages of the concepts and methods introduced. This approach was devised with the intent of enabling students to solve daily life problems as they go up the learning curve and for them to fully grasp the conceptual basis that will be built upon in subsequent grades.

After amalgamation of the efforts of experts and experienced authors, this book was reviewed and finalized after extensive reviews by professional educationists. Efforts were made to make the contents student friendly and to develop the concepts in interesting ways.

The National Book Foundation is always striving for improvement in the quality of its books. The present book features an improved design, better illustration and interesting activities relating to real life to make it attractive for young learners. However, there is always room for improvement and the suggestions and feedback of students, teachers and the community are most welcome for further enriching the subsequent editions of this book.

May Allah guide and help us (Ameen).

Dr. Raja Mazhar Hameed

Managing Director

National Book Foundation

Contents

Unit 1: Computer Systems

1.1 Brief History of Computer Systems.....7
1.2 Understanding Systems and their Types.....12
1.3 Core Components of a Computer System.....15
1.4 Von Neumann Architecture.....21
1.5 Data Transmission within a computer system.....23
1.6 Types and Hierarchy of Computer Memory.....25
1.7 Software Engineering and Hardware Engineering.....30
1.8 Computer Software.....31
1.9 Data Communication.....39

Unit 2: Computational Thinking & Algorithms

2.1 Problems.....66
2.2 Identifying a Computing Problem.....67
2.3 Problem Solving.....70
2.4 Input-Processing-Output (I-P-O) Model.....71
2.5 Input-Processing-Output (I-P-O) Chart.....71
2.6 Computational Thinking.....72
2.7 Principles of Computational Thinking.....75
2.8 Methods to Design a Solution.....76

Unit 3: Programing Fundamentals

3.1 Introduction.....88
3.2 HTML.....91
3.3 CASCADING STYLE SHEETS (CSS).....99
3.4 Java Script.....107
3.5 Debug The Code.....119
3.6 Create a Dynamic Website.....120

National Book Foundation

Unit 4: Data and Analysis

4.1 Data and Analysis:.....126
4.2 Data types in Data Science.....129
4.3 Big Data.....133

Unit 5: Application of Computer Science

5.1 Artificial Intelligence and Machine Learning 143
5.2. Different Areas of AI.....148
5.3 Applications of AI.....151
5.4 Cloud Computing.....156

Unit 6: Impacts of Computing

6.1 Safe and Responsible Use of Computer.....164
6.2 Responsible use of hardware and software.....166
6.3 Safe use of digital platform.....169
6.4 Laws to protect user privacy and intellectual property.....170
6.5: Computing innovation.....172
6.6 Malicious software and Key concept.....176
6.7 Information privacy, system security, and usability trade-off..177
6.8 Disinformation and Fake News.....178
6.9 Social Networking.....178
6.10 Approaches to computing effects.....179
6.11 Intellectual property protection key terms.....182

Unit 7: Entrepreneurship

7.1 YOU WILL BE YOUR OWN BOSS!.....190
7.2 COST OF PRODUCTION.....197
7.3 Devising a Business Plan.....198

National Book Foundation

UNIT 1

COMPUTER SYSTEMS

After completing this lesson, you will be able to:

- define and describe types of systems (artificial, natural), computer hardware components such as computer architecture (CPU, microprocessors, etc.)
- identify and explain system software, application software, lowlevel and high-level programming languages, and their uses.
- identify and analyze data communication, computer networks, networking devices, basic networking systems and understand how data is transmitted and key concepts such as protocols, speeds, etc.

Foundation

Introduction

A computer system is a fundamental and important part of modern life. It has revolutionized the way we work, communicate, learn, and entertain ourselves. A computer system is not just a single device but a sophisticated combination of hardware and software components that work together to process information, solve problems, and execute a multitude of tasks.

Understanding computer systems is very important in today's digital age, whether some one is a casual user or a professional in the field of computing. It empowers the users to utilize the capacities of computers for diverse purposes, from business and scientific research to creative activities and entertainment.

1.1 Brief History of Computer Systems and Generations of Computers

A computer is a programmable electronic device that performs arithmetic and logical operations automatically using a set of instructions provided by the user. When we study the aspects of computing and computers, it is important to know about the history of computers.

1.1.1 Early Computing Devices

Humans used sticks, leaves, stones and bones as counting tools before computers were invented. More computing devices were produced as technology advanced and the human intelligence improved over time. A few early-age computing devices are discussed as follows.

Abacus

Abacus was one of the earliest counting devices, consisting of beads or stones on rods or wires. It has been used for centuries in various cultures for arithmetic calculations by sliding beads to represent different numerical values. Abacus is shown in Fig. 1.1.

Fig.1.1 Abacus

Napier's Bone

John Napier developed Napier's bones, a manually operated calculating device. It used 9 separate strips (bones) marked with numerals to multiply and divide. It was also the first machine to calculate using the decimal point system.

Napier's Bone is shown in Fig. 1.2.

Fig.1.2 Napier's Bone

National Book Foundation

Pascaline

Pascaline was invented in 1642 by Bliaise Pascal, a French mathematician. It was thought to be the first mechanical and automated calculator. It consisted of a wooden box with gears and wheels in it. Pascaline is shown in Fig. 1.3.

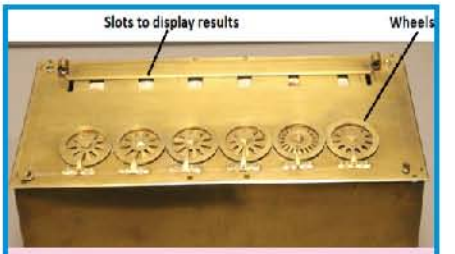


Fig.1.3 Pascaline

Stepped Reckoner or Leibniz wheel

Fig.1.4 Leibniz wheel

In 1673, a German mathematician named Wilhelm Leibniz improved on Pascal's invention to create this device. It was a digital mechanical calculator known as the stepped reckoner because it used grooved wheels instead of gears. Leibniz wheel is shown in Fig. 1.4.

Difference Engine

In the early 1820s, Charles Babbage created the Difference Engine. It was a mechanical computer that could do basic computations. It was a steam-powered calculating machine used to solve numerical problems.

Analytical Engine

Charles Babbage created another calculating machine, the Analytical Engine, in 1830. It was a mechanical computer that took input from punch cards. It was capable of solving any mathematical problem and storing data in memory. Analytical Engine is shown in Fig. 1.5.

Fig.1.5 Analytical Engine

Tabulating machine

Fig.1.6 Tabulating Machine

An American Statistician - Herman Hollerith invented this machine in the year 1890. Tabulating Machine was a punch card-based mechanical calculator. It could compute statistics and record data or information. Hollerith began manufacturing these machines in his company, which ultimately became International Business Machines (IBM) in 1924. Tabulating machine is shown in Fig. 1.6.

Differential Analyzer

Vannevar Bush introduced the first electrical computer, the Differential Analyzer, in 1930. This

Fig.1.7 Differential Analyzer

National Book Foundation

machine was made up of vacuum tubes used as switches to electrically impulse to do calculations. It was capable of performing 25 calculations per minute. Differential Analyzer is shown in Fig. 1.7.

Mark I

The next successful computing machine invented was a digital computer known as Mark-I. It was invented by Howard Aiken in 1944. Mark-I could add three numbers having eight digits in one second. It could print out its results on punched cards or on an electric typewriter. Mark-I was 50 feet long, 8 feet high and weighed about 5 tons. It used 3,000 electric switches. Mark-I is shown in Fig.1.8.

Fig.1.8 Mark-I Computer

1.1.2 Computer Generations

History of computers is a chain that runs from the ancient abacus and the analytical engine of the nineteenth century, through the modern quantum computers of present age. It is generally divided into five generations. Each generation of computers is characterized by major technological developments of that time.

First Generation Computers (1940 - 1956)

First-generation computers, emerged in the late 1940s and lasted through the early 1950s, were characterized by the use of vacuum tube technology. Vacuum tubes were used as a main electronic devices in the first generation computers. It consists of a glass tube containing electrodes (cathode, anode, and some additional elements) and a partial vacuum. A vacuum tube is shown is Fig.1-9.

Fig.1.9 Vacuum Tube

The following are some characteristics of first generation computers.

- Vacuum tubes were used in first-generation computers.
- The processing speed was slow.
- Memory capacity was limited.
- These computers were massive, occupying entire rooms.
- First-generation computers were both costly and unreliable.
- They consumed significant power and generated substantial heat.
- Input relied on punched cards.
- Output was obtained through printouts via electric typewriters.
- Machine language was the only programming paradigm.

Some examples of first generation Mini/Mainframe computers are ENIAC, UNIVAC I, IBM 604, Mark-I and EDSAC.

National Book Foundation

Second Generation Computers (1956 - 1963)

Second Generation computers emerged in the late 1950s and extended through the early 1960s. This period marked a significant advancement in computing technology, characterized by the transition from vacuum tubes to transistors. Transistor functions like a vacuum tube. It was faster, more reliable, smaller and much cheaper than vacuum tube. A transistor is shown in Fig 1-10.

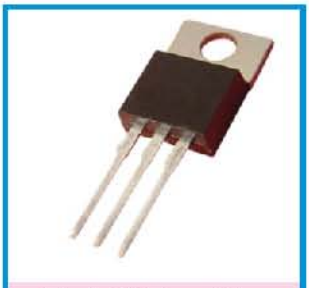


Fig.1.10 Transistor

The following are some characteristics of second generation computers.

- Second generation computers replaced vacuum tubes with transistors, leading to enhanced efficiency.
- The adoption of transistors resulted in a reduction in computer size, accompanied by improvements in speed and memory capacity.
- Second-generation computers demonstrated increased reliability and cost-effectiveness.
- Key input and output methods included punch card readers, magnetic tapes, magnetic disks, and printers.
- Assembly language was employed for programming purposes.
- This generation introduced high-level programming languages such as FORTRAN and COBOL.

Some examples of second-generation computers comprise UNIVAC II, IBM 7030, General Electric GE 635, and Control Data Corporation's CDC 1604 computers.

Third Generation Computers (1963 - 1971)

Third-generation computers emerged in the 1960s and extended into the 1970s. This era marked further advancements in computing technology, characterized by the use of integrated circuits (ICs) and the development of smaller, faster, and more reliable systems. IC chips are shown in Fig.1-11.

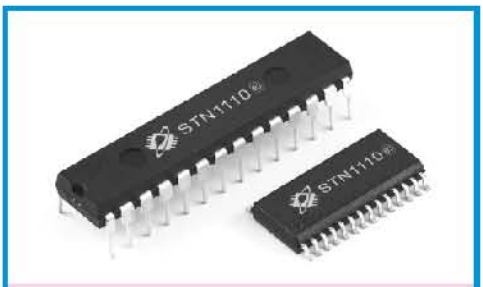


Fig.1.11 IC Chips

The following are the characteristics of third generation of computers.

- Third-generation computers used Integrated Circuit (IC) chips.
- The utilization of IC chips led to enhancements in computer speed and memory.
- These computers demonstrated improvements in energy efficiency, size reduction, cost-effectiveness, and reliability compared to second-generation computers.
- Interaction with third-generation computers involved the use of a keyboard and monitor.
- These computers had the capability to concurrently run multiple application programs.

DO YOU KNOW?
Intel invented the world's first microprocessor, the Intel 4004 in November, 1971

National Book Foundation

Examples of third-generation computers include IBM System/360 and Control Data Corporation's 3300 and 6600 computers.

Fourth Generation Computers (1971 - Present)

Fourth-generation computers, starting from late 70s to the present, are characterized by significant advancements in technology, particularly the development of Large Scale Integration (LSI) and Very Large Scale Integration (VLSI) chips. One of the key innovations of this era was the development of the microprocessor, a single chip capable of handling all processing tasks within a computer. A microprocessor is shown in Fig.1-12.

Fig.1.12 Microprocessor

The following are the characteristics of fourth generation of computers.

- The introduction of microprocessors marked a defining feature of fourth-generation computers, leading to the emergence of microcomputers.
- Fourth-generation computers are known for their exceptional speed, large storage capacity, and the incorporation of advanced input/output devices.
- Microcomputers in this generation are characterized by their small size, high reliability, low power consumption, and affordability.
- A wide variety of software became available for use in microcomputers during the fourth generation.
- Operating systems with Graphical User Interfaces (GUIs) were developed during this period, enhancing user interaction and experience.
- Fourth-generation computers support multimedia software, enabling the integration of text, image, sound, and video.
- These computers are compatible with modern programming languages such as Visual Basic, C++, Java, and Python, facilitating the development of powerful software applications.
- Fourth-generation computers support a diverse range of portable and wireless input/output devices.

Examples of microprocessors developed during this era include the Intel Pentium series, Dual Core, Core2 Duo, Core i3, i5, i7, and AMD Athlon. Notable fourth-generation computer models include the IBM ThinkPad series, HP Pavilion series, Dell Inspiron series, as well as Apple's MacBook Pro and MacBook Air series.

Fifth Generation Computers

The timeline for the fifth generation is not as precisely defined as the earlier generations, but it is generally associated with ongoing advancements in computing that were expected to emerge in the late 20th century and beyond. The main objective of fifth generation of computers is to develop devices that can understand natural languages and have thinking power. This is a big challenge for computer developers and programmers to design such systems and software for them.

National Book Foundation

The following are the characteristics of fifth generation of computers.

- The primary focus of fifth-generation computers tis to develop and utilize **AI (Artificial Intelligence)** technologies. This involves machines (called **Robots**) with the capability to learn, think, innovate, reason, and solve problems automatically and independently.
- Fifth-generation computers support advanced parallel processing capabilities, allowing them to execute multiple tasks simultaneously and handle complex computations more efficiently.
- A key aspect of fifth-generation computing is the ability to understand and respond to human languages. This involves developing systems capable of **NLP (Natural language processing)** and communication.
- These computers are planned to incorporate advanced **ES (Expert systems)**, which are software programs designed to replicate the decision-making abilities of a human experts in specific fields, like medical, mining and engineering.
- In this generation of computers user interfaces have become more intuitive and user-friendly, incorporating features like **Voice recognition** and gesture-based controls.

1.2 Understanding Systems and their Types

What is a System?

A "**System**" refers to a collection of interconnected or interrelated components or elements that work together to achieve a specific purpose or function. Systems can be found in various aspects of life, from **natural** ecosystems to human-made **artificial** systems. Any kind of system is a group of resources which work collectively in order to produce desired results from given inputs. It accepts input and produces the output.

Understanding and categorizing systems is important for various fields, including engineering, biology, sociology, and management, as it allows for better analysis, design, and optimization of these systems to achieve their intended goals. Different types of systems may show distinct properties and behaviors, which necessitate different approaches to study and manage them effectively.

1.2.1 Natural and Artificial Systems

Natural Systems

A natural system is an interconnected collection of elements or components that exist in the nature. These systems are typically found in the environment and are characterized by their ability to self-regulate, adapt, and maintain a certain degree of stability. Natural systems are incredibly diverse and fascinating, each with its unique characteristics and interactions. Some natural systems are shown in Fig. 1.13.

Teacher's Guide

- Encourage students to explore additional resources, such as books, websites, and educational videos, to deepen their understanding of computing history and systems.
- Stay updated on emerging trends and advancements in computing to inspire ongoing curiosity and learning among students.

National Book Foundation

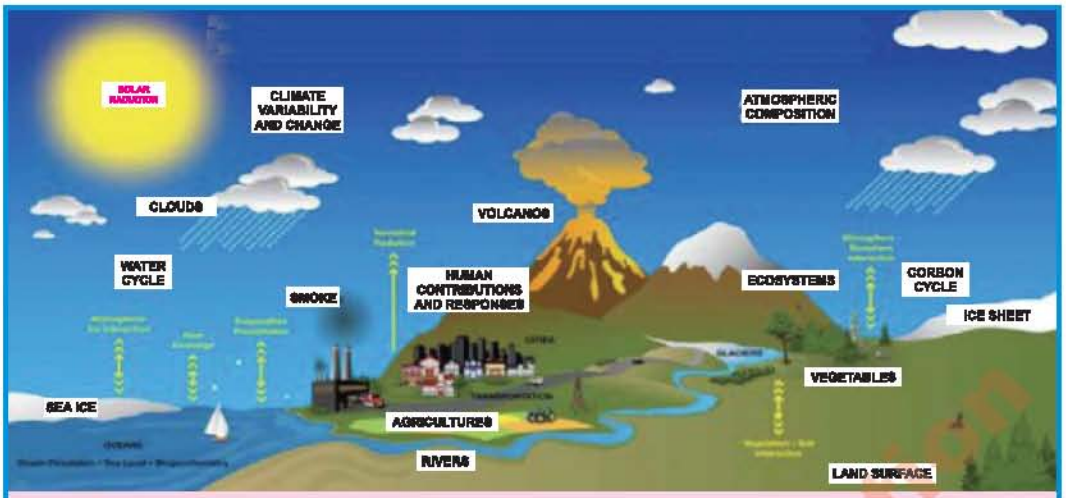


Fig.1.13 Natural Systems

The following are a few examples of natural systems.

Ecosystems: Ecosystems are perhaps the most common and diverse natural systems. They encompass various types, including:

- **Forest Ecosystems:** Such as a tropical rainforest with its myriad of plant and animal species.
- **Aquatic Ecosystems:** Like freshwater lakes, rivers, and marine ecosystems such as coral reefs.
- **Grassland Ecosystems:** Such as the African savanna, home to grazing animals like zebras and wildebeests.

Weather Systems: Weather systems involve the interactions of the Earth's atmosphere, including phenomena like rainfall, wind patterns, and temperature changes.

Geological Systems: These systems encompass geological processes and features like:

- **Plate Tectonics:** The movement of Earth's lithospheric plates, leading to phenomena like earthquakes and volcanic eruptions.
- **Mountain Systems:** Such as the Himalayas, formed by tectonic plate collision.

Hydrological Systems: These systems involve the movement, distribution, and quality of water on Earth, including rivers, lakes, and the water cycle.

Solar System: Our solar system itself is a natural system, with the Sun, planets, moons, asteroids, and comets all interacting under the influence of gravity.

Biological Systems: Biological systems encompass a wide range of living organisms and their interactions:

- **Human Body:** A complex biological system with organs, tissues, cells, and biochemical processes.
- **Coral Reef:** An ecosystem built by coral colonies and inhabited by various marine species.

National Book Foundation

- **Rainforest Canopy:** The upper layer of a rainforest, home to a unique set of plants and animals.

Artificial Systems

Artificial systems, also known as man-made or human-made systems, are created and designed by humans to serve specific purposes, solve problems, or achieve particular goals. Unlike natural systems, which occur organically in the natural world, artificial systems are intentionally constructed by humans to address various needs and objectives. These systems can range from simple devices to highly complex structures, and they exist in numerous domains.

Artificial systems are essential components of modern society, contributing to our ability to meet a wide range of needs and advance in various fields. They often require careful planning, engineering, and maintenance to function efficiently and effectively. Some artificial systems are shown in Fig. 1.14.

Fig. 1.14 Artificial Systems

The following are some common examples of artificial systems.

Communication Systems:

- **Telephone Networks:** Created to facilitate voice communication over long distances.
- **Internet and Computer Networks:** Built to enable data sharing and digital communication globally.
- **Satellite Communication Systems:** Developed for long-distance, wireless communication.

Information Systems:

- **Databases:** Used for storing and retrieving data efficiently.
- **Software Applications:** Such as word processors, spreadsheets, and video editing software.

Transportation Systems:

- **Automobiles:** Designed for personal and mass transportation on roads.
- **Aircraft:** Engineered for air travel, including commercial airplanes, helicopters, and drones.
- **Trains and Rail Systems:** Developed for efficient land transportation on tracks.
- **Subways and Mass Transit:** Designed to move large numbers of people within urban areas.

National Book Foundation

Energy Systems:

- **Power Plants:** Designed to generate electricity using various energy sources like coal, natural gas, nuclear, or renewable resources.
- **Renewable Energy Systems:** Including solar panels, wind turbines, and hydroelectric plants.
- **Electrical Grids:** Infrastructure for the distribution of electrical power.

Manufacturing Systems:

- **Factory Automation:** Systems that automate manufacturing processes, such as robotics and conveyor systems.
- **Assembly Lines:** Organized systems for mass-producing goods.

Healthcare Systems:

- **Hospital Information Systems (HIS):** Designed to manage patient records, billing, and other healthcare data.
- **Medical Devices:** Including MRI machines, X-ray equipment, and artificial organs.

1.3 Core Components of a Computer System

A computer system comprises of several core components that work together to perform various tasks. The essential components of a computer system include Input devices, Output devices, System unit (motherboard, memory, CPU, power supply, etc.), and data storage devices.

1.3.1 Input Devices

Input devices are used to provide data into the computer system. Input devices allow us to communicate with the computer. Some commonly used input devices are keyboard, mouse, microphone, scanner, barcode reader, digital camera and touch screens.

Keyboard

It is the main input device to communicate with the computer. It allows the computer user to enter letters, numbers and special symbols into the computer. A keyboard is shown is Fig.1.15.

POINT TO PONDER:
Why the keys on keyboard are not arranged in alphabetical order?

Fig.1.15 A standard keyboard

Teacher's Guide

- Explore the characteristics and behaviors of natural systems like ecosystems and weather patterns.
- Discuss the design, purpose, and functionality of artificial systems such as communication networks, transportation systems, and healthcare systems.

National Book Foundation

It is a hand-held device used to control the movement of cursor or pointer on the screen. It has two or three buttons at the front that allows the computer user to make selection in menu, draw graphics or open files, folders and programs. A typical mouse is shown is Fig.1.16.

Fig.1.16 Mouse

Microphone

It is a device that allows computer user to input audio into the computer. It changes audio signals into electrical signals which are translated into digital form by the sound card for processing in the computer. A microphone is shown in Fig.1.17.

Fig.1.17 Microphone

Scanner

It is a device that captures images from photographs, magazines, books etc. and stores them in computer in digital form. These images can be edited, displayed on the screen or inserted in documents. A scanner is shown in Fig.1.18.

Fig.1.18 Scanner

Barcode Reader

Fig.1.19 Barcode Reader

It is a device that reads the barcode printed on products that represents product code, description and price. This information is used by the computer to print bill for the customer. A barcode reader is shown in Fig.1.19.

Digital Camera

Fig.1.20 Digital Camera

It is input/output device used to capture pictures and store them in digital form. These pictures can be downloaded to computer for editing, viewing or inserting in documents. A digital camera is shown in Fig.1.20.

Touch Screen

Fig.1.21 Touch Screen

It is a pressure-sensitive display screen that is used to interact with the computer by touching pictures or words with finger. Touch screen is more commonly used with mobile phone and tablet. A touch screen is shown in Fig.1.21.

1.3.2 System Unit

System unit is the main part of computer. It includes motherboard, power supply and drives (such as DVD and hard disk) inside the computer casing. All the input/output devices of a computer are connected to system unit through the ports.

Motherboard

Motherboard is the main circuit board inside the system unit. It contains microprocessor, main memory, expansion cards, many IC chips, connectors and other electronic components. It has many buses (electric pathways) printed on it. These are

National Book Foundation

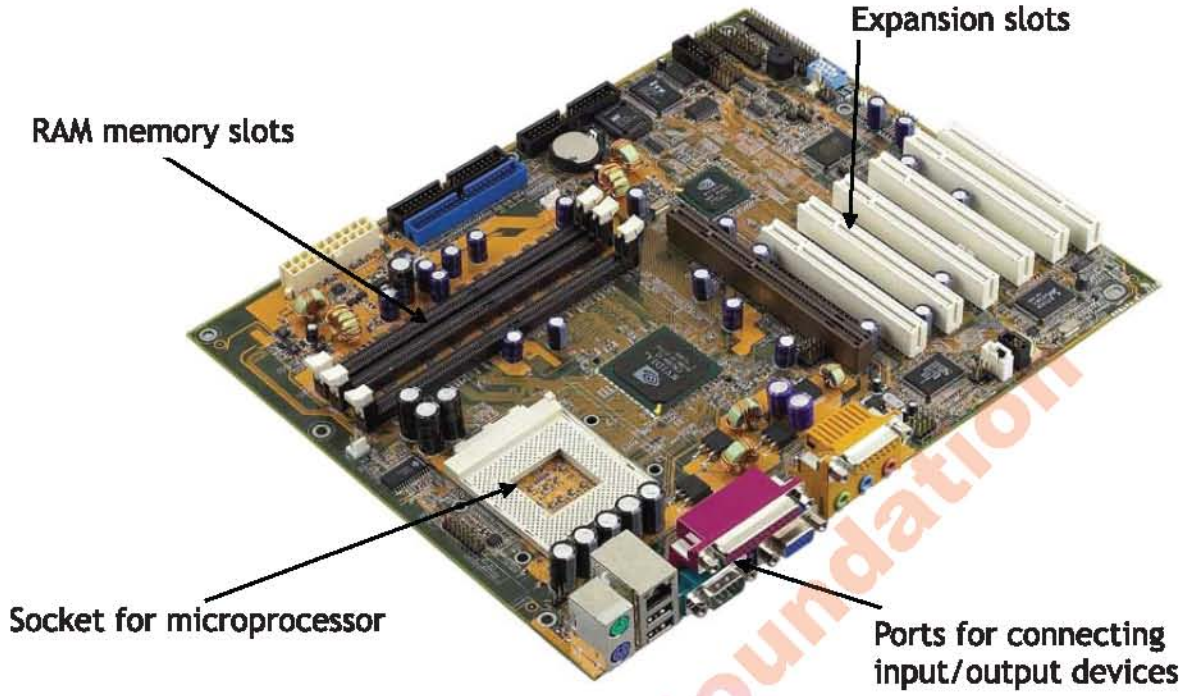


Fig.1.22 Motherboard

used to transmit information between various components of the computer. All the input/output devices are connected to the motherboard. A motherboard is shown in Fig.1.22.

Microprocessor

A microprocessor is the main chip on the motherboard that controls all the activities of the computer. It is also known as Central Processing Unit (CPU) or simply processor. It contains Control Unit (CU), Arithmetic Logic Unit (ALU) and Registers. A microprocessor and the block diagram of CPU are shown in Fig.1.23.

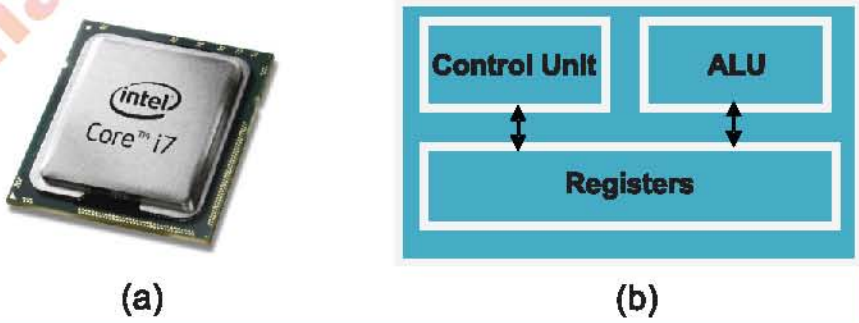


Fig.1.23 (a) Microprocessor (b) Block diagram of microprocessor

1.3.3 Storage Devices

Storage devices are used to store programs and data that are not currently used by

National Book Foundation

the computer. They have huge storage capacity. Therefore, they are also known as mass storage devices or secondary memory. Hard disk is the most commonly used storage device that is fixed inside the system unit. Portable storage devices are CD, DVD, memory cards and USB flash drive. Portable storage devices have less storage capacity than hard disk but they are cheap and easy to carry.

Hard Disk

A hard disk is a magnetic storage device used to store computer data permanently. It has storage capacity of hundreds of Gigabyte (GB). It is fixed inside the computer casing. Portable hard disk is also available that is attached to USB port.

Compact Disk (CD)

CD (DVD) is a portable optical storage device with a storage capacity of 700 Megabytes (MB). A CD is 1.2 millimeter thick with a diameter if 120 millimeters. CD drive is used to read data from or write data to a CD.

Digital Versatile Disk

DVD is also portable optical storage device. It has the same thickness and diameter as CD but has more storage capacity. Its storage capacity is in the range of 4 to 16 GB. A DVD writer is installed in the computer to read data from or write data to a DVD. A CD can also be used in a DVD writer.

Memory Card

Memory card is a small storage device having storage capacity of few Gigabytes. It is available in different sizes and storage capacities. Memory cards are generally used in laptop computers and portable devices such as mobile phone and digital camera for storing pictures, audio and video. A memory card is shown in Fig.1.24.

Fig.1.24 Memory Card

USB Flash Drive

USB flash drive is a small portable drive that is connected to computer through USB port. It is also known as USB memory. It is very fast in operation and its storage capacity is up to 128 GB till now. A USB flash drive is shown in Fig.1.25.

Fig.1.25 USB Flash Drive

1.3.4 Output Devices

Output devices are used to display text, graphics, and images on the monitor or to print information on paper. Information displayed on monitor is known as softcopy and anything printed on paper is known as hardcopy or printout. Commonly used output devices are monitor, printer, plotter and speaker.

Monitor

It is an output device that has a screen on which information is displayed. It has

National Book Foundation

two common types i.e. CRT (Cathode Ray Tube) monitor and LED (Light Emitting Diode) monitor. CRT monitor is very similar to old television. It is almost obsolete due to its big size and low display quality. LED monitor is slim, uses less power and has better display quality than CRT monitor. CRT and LED monitors are shown in Fig.1-26.

Fig.1.26 (a) CRT Monitor (b) LED Monitor

Printer

Printer is an output device that prints text and graphics on paper which is known as hardcopy. There are two types of printers which are impact and non-impact printers.

FOR YOUR INFORMATION:
The first high-speed printer was developed in 1953 by Remington Rand (an early American business machines manufacturer) for use on UNIVAC computer.

Impact printer

Impact printer uses electro-mechanical mechanism which causes the character shape to strike against the paper and leave an image of the character on the paper. Dot matrix printer is the most commonly used impact printer. The printing speed varies from 50 to 500 cps (characters per second). Their printing is very cheap but print quality is poor. They produce lot of noise while printing. These printers are still in use for printing invoices, bank statements, utility bills, etc. A Dot matrix printer is shown in Fig.1-27(a).

(a) (b) (c)

Fig.1.27 (a) Dot Matrix Printer (b) Inkjet Printer (c) Laser Printer

Non-Impact printer

Non-Impact printer prints without striking the paper. There are two types of non-Impact printers which are inkjet and laser printers. Inkjet printer stores ink in cartridge and sprays on paper through fine nozzles on the print-head. Laser printer uses technology similar to photocopying machine. Laser printer is more expensive, faster and has very high print quality compared to inkjet printer. Inkjet printers are used in all sectors such as homes and simple businesses. Laser printers are perfect for large scale businesses. Inkjet and laser printers are shown in Fig.1.27.(b,c).

National Book Foundation

Plotter

Plotter is an output device used for printing engineering drawings, machine parts, building designs, maps, charts and panna-flexes etc. on large size papers/sheets. Such large size printing is not possible on printers. It is more expensive than printer. There are two types of plotters, that is, ink plotter and pen plotter. Ink plotter is used for printing images whereas pen plotter is used for printing engineering drawings, machine parts, building designs, etc. Plotter is a slow output device but its printing quality is good. A plotter is shown in Fig.1.28.

Fig.1.28 Plotter

Speaker

Speaker is a device used to produce audio output. A pair of speakers are attached to the sound card on the motherboard. Speakers are commonly used with multimedia software and for playing music and videos on computer. A pair of speakers are shown in Fig.1.29.

Fig.1.29 Speakers

1.3.5 Ports, Expansion Slots and Expansion Cards

Ports

Port is an interface for connecting various devices to the system unit. These are located on the motherboard and are usually seen at the back of the system unit. There are various types of ports for connecting keyboard, mouse, monitor, microphone, speakers and other input/output devices as shown in Fig.1.30. (a). In modern computers, USB (Universal Serial Bus), HDMI (High Definition Multimedia Interface), DVI (Digital Visual Interface), Audio and LAN (Local Area Network) ports are used for connecting various devices to the computer. These devices include digital camera, scanner, printer, external hard disk or DVD writer and USB memory, etc.

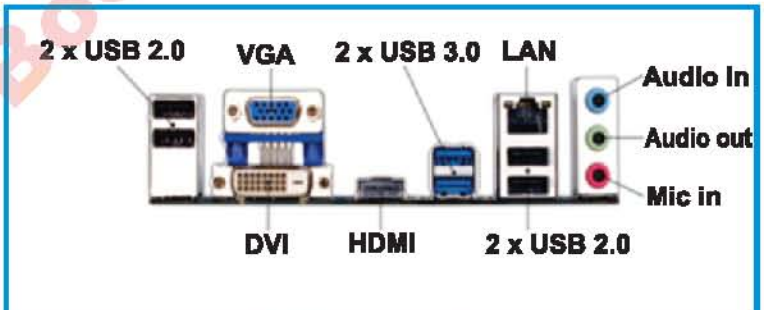


Fig.1.30 Ports on motherboard

Expansion Slots and Expansion Cards

Expansion slots are long narrow sockets on the motherboard used for installing expansion cards.

Fig.1.31 Network card

National Book Foundation

Expansion cards are small circuit boards. These cards add new capabilities to the computers. Commonly used expansion cards are sound card, graphics card, modem card and network card. In modern computers these cards are built-in on the motherboard. A network card is shown in Fig.1.31.

1.4 Von Neumann Architecture

Von Neumann Architecture is an essential concept in computer science that explains how a computer's hardware and software work together to process information. It was first published by John von Neumann in 1945. His computer architecture design consists of a Control Unit, Arithmetic and Logic Unit (ALU), Memory Unit, Registers and Inputs/Outputs.

Von Neumann architecture is based on the stored-program computer concept, where instruction data and program data are stored in the same memory. This design is still used in most computers produced today. Modern Von Neumann architecture is shown in Fig. 1.32.

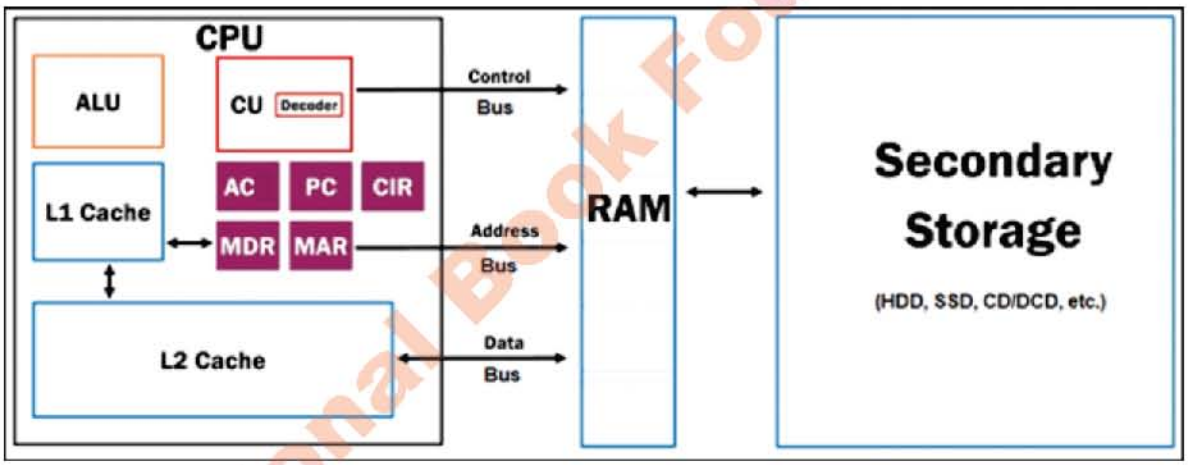


Fig.1.32 Von Neumann architecture

National Book Foundation

Central Processing Unit (CPU)

The Central Processing Unit (CPU) is the main electronic circuit responsible for executing the instructions of a computer program. The CPU contains the ALU, CU and a variety of registers.

Registers

Registers are high speed storage areas in the CPU. All data must be stored in a register before it can be processed.

MAR	Memory Address Register	Holds the memory location of data that needs to be accessed
MDR	Memory Data Register	Holds data that is being transferred to or from memory
AC	Accumulator	Where intermediate arithmetic and logic results are stored
PC	Program Counter	Contains the memory address of the next instruction to be executed
CIR	Current Instruction Register	Contains the current instruction during processing

Arithmetic and Logic Unit (ALU)

The ALU allows arithmetical (add, subtract etc.) and logical (AND, OR, NOT etc.) operations to be carried out.

Control Unit (CU)

The control unit controls the operation of the computer's ALU, memory and input/output devices, telling them how to respond to the program instructions interpreted from the memory unit. The control unit also provides the timing and control signals required by other computer components.

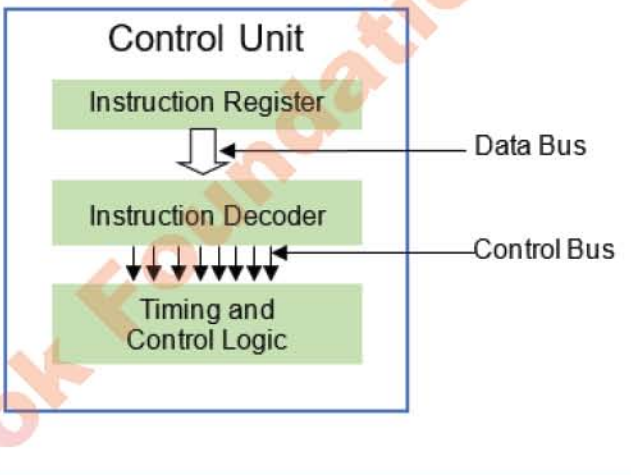


Fig.1.33 Control Unit

Buses

Buses are the pathways or lines by which data is transmitted from one part of a computer to another, connecting all major internal components to the CPU and memory. A standard CPU system bus is comprised of a control bus, data bus and address bus.

Address Bus	Carries the addresses of data (but not the data) between the processor and memory
Data Bus	Carries data between the processor, the memory unit and the input/output devices
Control Bus	Carries control signals or commands from the CPU in order to control and coordinate all the activities within the computer

National Book Foundation

Memory Unit

In Von Neumann architecture the memory unit consists of RAM and Cache memory, sometimes referred to as primary or main memory. This memory is fast and also directly accessible by the CPU. RAM is split into partitions. Each partition consists of an address and its contents (both in binary form). The address uniquely identifies every location in the memory.

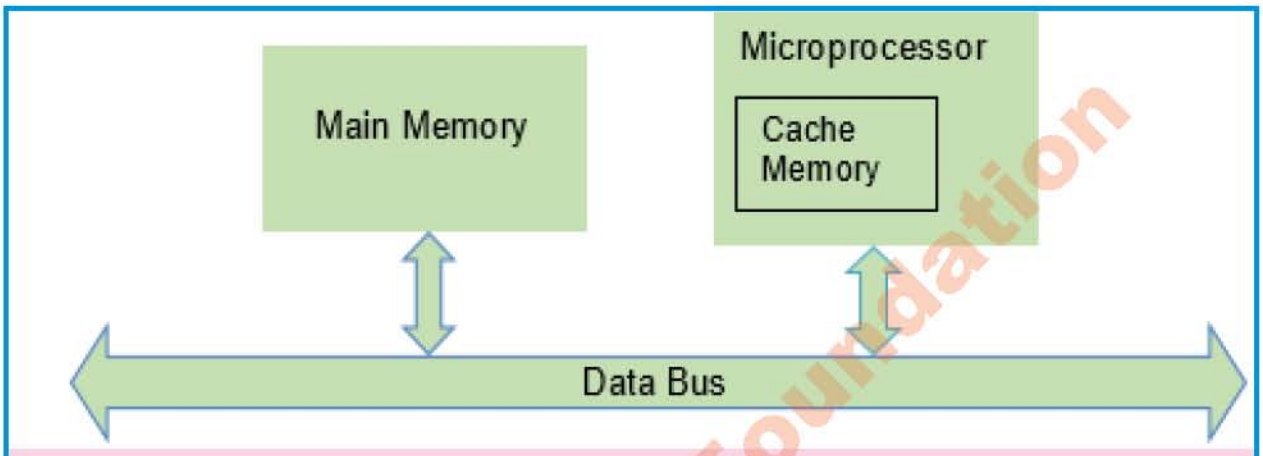


Fig.1.34 Cache Memory

Input/Output (I/O) Controller

This component manages the flow of data between the CPU and external devices like hard drives, USB devices, and network interfaces.

1.5 Data Transmission within a computer system

Data transmission within a computer system involves the movement of data/information between various components such as the CPU, memory, storage devices, and input/output devices. This process is vital for the proper functioning of a computer and its ability to execute tasks efficiently.

The following points give an overview of how data is transmitted within a computer system:

Bus Architecture: Computers use a bus architecture to transmit data. A bus is a communication pathway that allows the transfer of data and control signals between various components such as the CPU, memory, and peripheral devices. It is just like a highway system for data inside the computer system.

Teacher's Guide

- Organize interactive demonstrations and hands-on activities to help students understand the functionality of computer components.
- Develop assessments that evaluate students' comprehension of core concepts, including quizzes, projects, and presentations.
- Provide opportunities for practical tasks and demonstrations to assess students' ability to apply their knowledge.

National Book Foundation

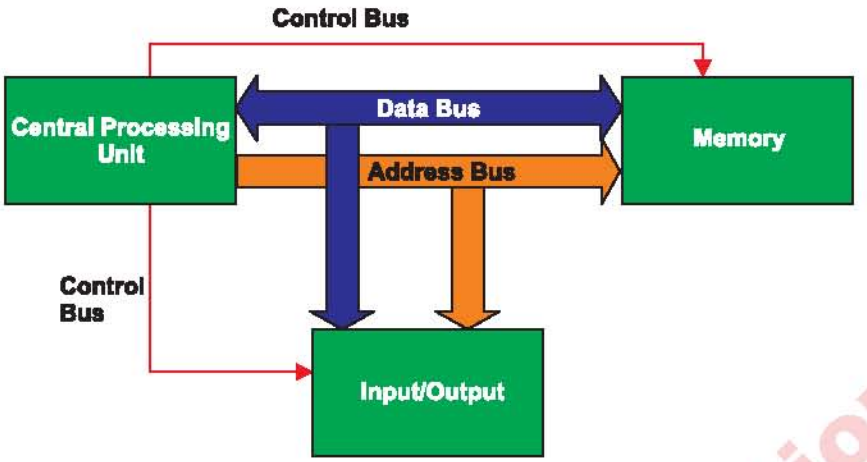


Fig.1.35 System Buses

Data Paths: Inside the CPU, data paths are dedicated paths known as data circuits that facilitate the movement of data between various functional units. These functional units include the Arithmetic Logic Unit (ALU), registers, cache, and other components involved in data processing. The data paths allow the CPU to perform operations on data by providing routes for data to move within the processor.

Data paths are more internal and pertain to how data moves within the CPU, while bus architecture addresses the broader communication infrastructure that enables data transfer between the CPU and other parts of the computer system.

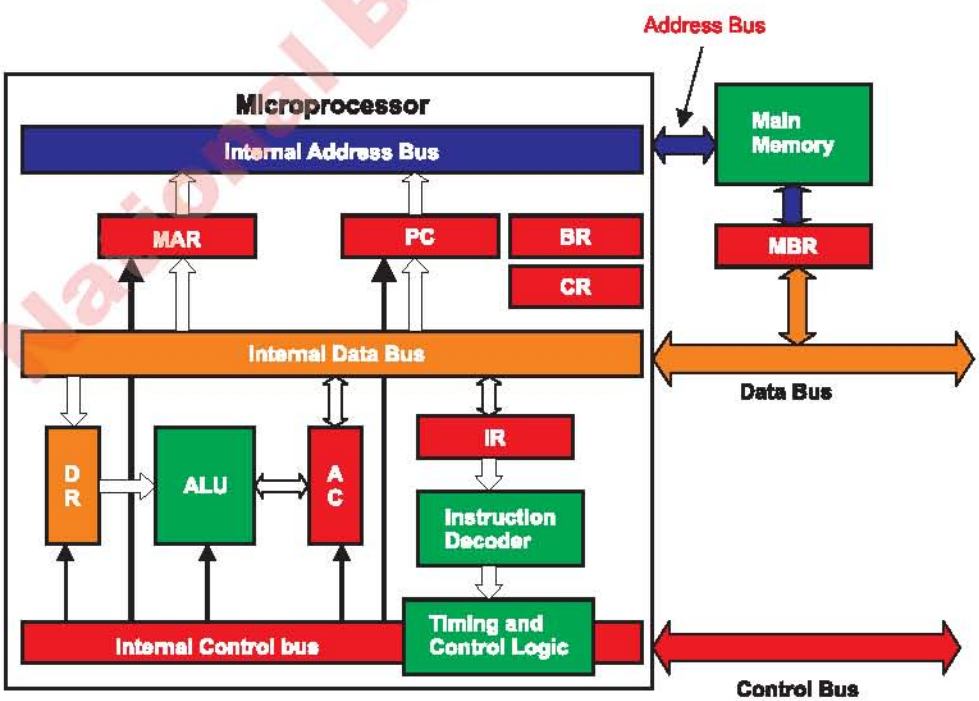


Fig.1.36 Microprocessor bus architecture with associated registers

National Book Foundation

Registers: Registers are small, high-speed storage units located within the CPU. They hold data that is frequently used by the CPU during processing. Data is quickly transferred between registers and main memory via the buses.

Memory Hierarchy: Modern computers use a memory hierarchy to improve data transmission. Data is stored in different levels of memory, ranging from high-speed but small Cache memory to larger and slower RAM, and even slower secondary storage (hard drives, SSDs). The CPU fetches data from the higher levels of the hierarchy first due to their faster access times.

Instruction Cycle: When a program is executed, it goes through a series of steps called the instruction cycle. This cycle involves fetching the next instruction from memory, decoding it to understand what operation is required, fetching operands from memory or registers, executing the operation, and storing the results back in memory or registers. It is called fetch-decode-execute cycle.

Pipeline Processing: Many modern CPUs use pipelining to increase efficiency. In a pipeline, multiple instructions are in different stages of execution simultaneously. This allows for better utilization of the CPU's resources and faster execution of instructions.

Interrupts and I/O: Input/output devices (e.g., keyboard, mouse, display, network interface) communicate with the CPU using interrupts. An interrupt is a signal that halts the current program's execution to handle an important event. This allows data to be transmitted between the CPU and these devices.

Parallelism: Some computer architectures use parallelism to improve data transmission speed. This can involve multiple cores within a CPU or even distributed systems with multiple interconnected computers working together.

1.6 Types and Hierarchy of Computer Memory

In computing memory refers to the physical devices used to store programs (sequence of instructions) or data on a temporary or permanent basis for use in a computer or other digital/computing devices. Memory in a digital computer contains the main part of operating system and all the application programs and related data that is being used. Fig. 1.37 shows different types of computer memory.

National Book Foundation

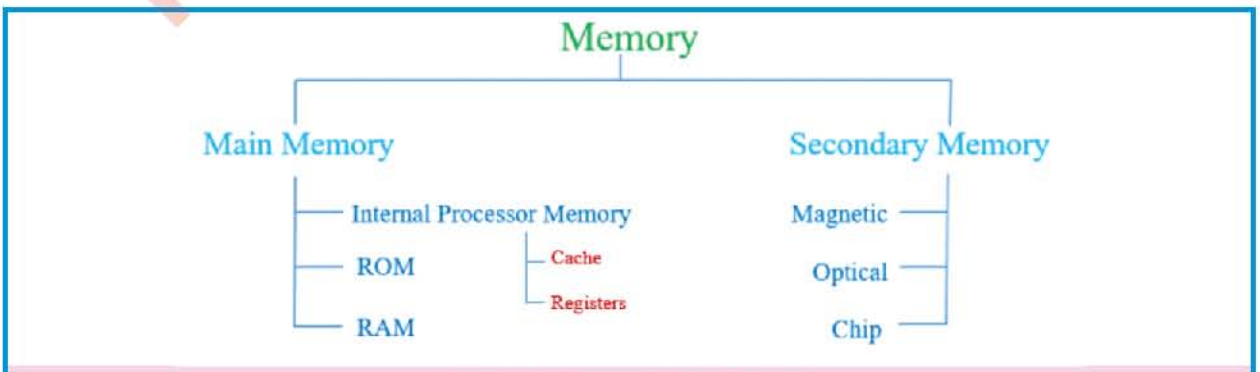


Fig. 1.37 Memory Classification

1.6.1 Memory Terminology

The following are some important memory terms.

Bit: The smallest unit of memory in digital computer is a bit, which stands for binary digit 0 or 1. The memory of a computer consists of millions of memory (or electronic) cells. Each cell contains one bit of information. The memory cell has two states, ON and OFF. The ON state represents a binary 1 and OFF state binary 0.

Byte: Byte is the basic unit of computer memory and it is the minimum piece of data to be processed by a computer. A group of 8 bits is known as one byte. One byte of memory is required to store one character in the computer, for example 'A', 'a', 'b', '*', etc. A byte is generally used to express the memory size of a computer. Computer memory is measured in terms of bytes. The higher units are Kilobyte (KB), Megabyte (MB), Gigabyte (GB) and Terabyte (TB). In future, memories will also be available in Petabyte (PB) and Exabyte (EB) as indicated in red colour in Table 1.1. The relationship between the memory units is shown in Table 1.1.

MEMORY UNIT	EQUIVALENT TO	
1 Byte	8 Bits	
1 Kilobyte (KB)	2^{10} Bytes	= 1024 Bytes
1 Megabyte (MB)	2^{20} Bytes	= 1024 KB
1 Gigabyte (GB)	2^{30} Bytes	= 1024 MB
1 Terabyte (TB)	2^{40} Bytes	= 1024 GB
1 Petabyte (PB)	2^{50} Bytes	= 1024 TB
1 Exabyte	2^{60} Bytes	= 1024 PB

Table 1.1 Memory Units and their Equivalents

Memory Word: In computing, the smallest amount or size of data that a computer can process is called memory word. It is a fixed-sized piece of data handled as a unit by the processor. The number of bits in a word is called the word size. Word size in modern computers typically ranges from 16 to 64 bits, depending on the size of the computer. A computer that has a bigger word size can transfer more bits into the microprocessor at a time for processing and this improves the processing speed of the computer.

Word Size: Word size refers to the number of bits that a computer's CPU can process or manipulate in a single instruction or operation. The word size of a CPU is a fundamental characteristic that affects its performance and capabilities. For example, a CPU with a 32-bit word size can process data in 32-bit chunks, while a CPU with a 64-bit word size can process data in 64-bit chunks. A larger word size generally allows a CPU to handle larger integers, perform more complex arithmetic operations, and address larger memory spaces.

1.6.2 Memory Built-up and Retention power

All types of computer memories, as for as their built-up or manufacturing is concerned, are divided into Chip memory, Magnetic memory and Optical memory. And as

National Book Foundation

far as their retention power is concerned these memories are divided into Volatile memory and Non-Volatile memory.

Fig.1.38 Chip Memory devices

Chip Memory

Chip is a small piece of semi-conducting material (usually silicon). A small circuit called IC (Integrated Circuit) is embedded on it. A typical chip contains millions of electronic components (transistors).

Chip memories are better in speed compared to other memory types due to the absence of mechanical moving parts. Unlike traditional forms of memory, chip memories rely on electric currents for their operation. This reliance on electrical processes contributes to their rapid data access and retrieval capabilities.

Examples of chip memory are main memory (RAM, ROM and Cache), Flash memory drives, memory cards, registers and Solid State drives(SSDs). Many special-purpose chips, known as application-specific integrated circuits, are also being made today for automobiles, home appliances, telephones, and other devices. Different types of chip memory devices are shown in Fig.1.38.

Magnetic Memory

DO YOU KNOW?
In September 1956 IBM launched the 305 RAMAC, the first 'SUPER' computer with a hard disk drive (HDD). The HDD weighed over a ton and stored 5 MB of data.

One of the most widely used types of digital data storage is magnetic memory/storage. This refers to any type of data storage using a magnetized medium. Magnetic tapes and disks are examples of magnetic memory devices. A thin layer of magnetic material is coated on the surface of magnetic tape and magnetic disks. Binary information is stored in the form of tiny magnetized and non-magnetized spots on the surface of magnetic tape or disk. A magnetized spot represents a binary 1 and a non-magnetized spot a binary 0. A read-write head moves very close to the magnetic surface. The head is able to detect and modify the magnetization of the material. Magnetic storage is widely used because it is relatively cheap in comparison with other storage technologies. The storage capacity is also very large, making it attractive for storing very large amounts of data. The major limitation of magnetic storage is that accessing the data can be quite slow. Hard disk is the common example of magnetic memory as shown in Fig 1.39.

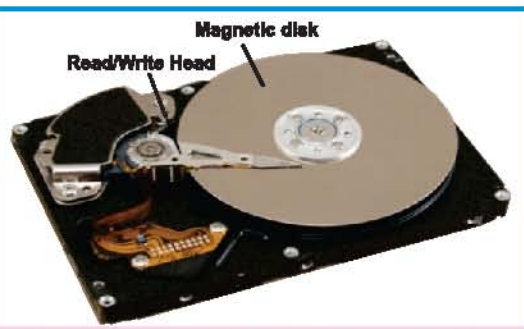


Fig.1.39 Magnetic disk with read/write

National Book Foundation

Optical Memory

In optical-storage technology, a laser beam encodes digital data onto an optical disk in the form of tiny pits and lands arranged in concentric tracks on the disk's surface as shown in Fig. 1.40. A low-power laser scanner is used to "read" data or information from these pits and lands, and converts it to digital form.

Optical storage provides cheaper and greater memory capacity than magnetic storage. An entire set of encyclopedias, for example, can be stored on a standard 12-centimetre (4.72-inch) optical disk. Optical disks include CDs, DVDs and Blu-ray disks(BDs)

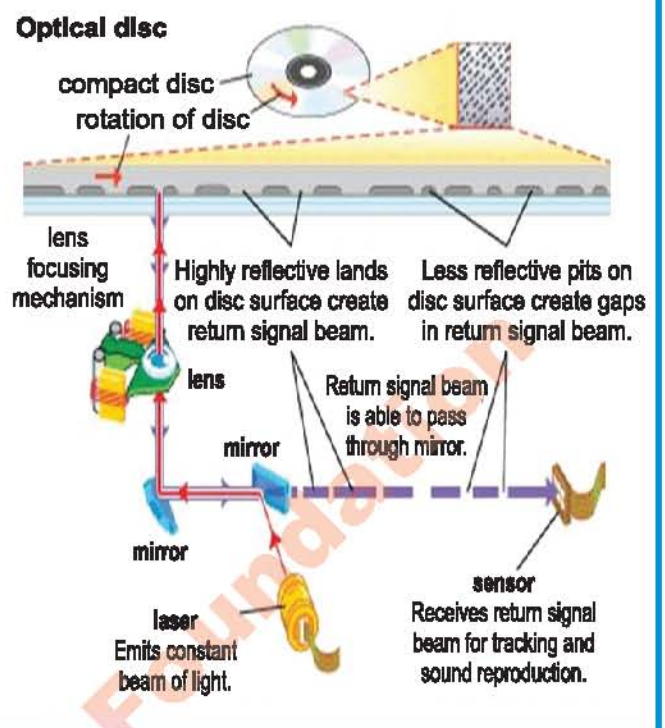


Fig.1.40 Optical Memory technology

1.6.3 Main Memory

Main Memory stores data and programs that are being executed by the computer. It also stores the results produced by the ALU after processing the data. There are three types of main memories on the motherboard which are ROM (Read Only Memory), RAM (Random Access Memory) and Cache. These are known as main memory or primary memory of computer.

ROM (Read Only Memory)

ROM is a single IC chip which is installed on the motherboard as shown in Fig.1.41.

Fig.1.41 ROM chip

It stores the Basic Input/Output System (BIOS) of computer that controls input/output devices and the start-up or boot process. BIOS programs test the computer's components when it is turned on and then load the operating system into the RAM to make the computer ready for operation.

BIOS programs are permanently stored in ROM when it is manufactured. It is non-volatile memory, that is, the programs stored in it are not lost when the computer is turned off. There are three common types of ROM which are PROM (Programmable ROM), EPROM (Erasable Programmable ROM) and EEPROM (Electronically Erasable Programmable ROM).

National Book Foundation

RAM (Random Access Memory)

RAM is high speed memory installed on the motherboard. It is READ/WRITE memory. Information can be read from or written into it. Programs are loaded into RAM from secondary storage devices such as hard disk or USB flash drive for execution by the microprocessor. It is volatile memory which means information stored in it, is lost when the computer is turned off.

Fig.1.42 RAM modules

RAM modules are installed in the memory slots on the motherboard. RAM modules are shown in Fig.1.42.

Cache Memory

Cache is a very small amount of extremely fast memory inside the microprocessor or on the motherboard. It is faster and more expensive than RAM. It stores information that is most frequently used by the computer. The purpose of using cache is to improve the processing speed of computer.

There are three types of cache memories which are Level 1 (L1), Level 2 (L2) and Level 3 (L3) as shown in Fig.1.43. L1 cache is built inside the microprocessor whereas L2 and L3 are on the motherboard. L1 cache is faster than L2 and L3 cache.

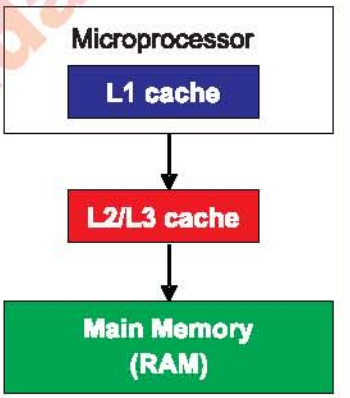


Fig.1.43 L1, L2 and L3 Cache Memories

1.6.4 Volatile and Non-Volatile Memory

Memory, on the basis of retention power, can be divided into two types i.e. volatile and non-volatile memory.

Volatile memory

Volatile memory is a temporary memory, that requires power (electricity) to maintain the stored information. Volatile memory retains the information as long as power supply is on, but when power supply is off or interrupted the stored memory is lost. It is also known as temporary memory. Examples of such memory are RAM (Random access memory), Cache memory and Registers.

Non-Volatile memory

Non-volatile memory is a permanent memory, that can retain the stored information even when powered off. Examples of non-volatile memory include ROM (Read-only memory), flash memory, magnetic storage devices (e.g. hard disks and magnetic taps), optical disks, and blue-ray disk. Non-volatile memory is typically used as secondary storage for long-term or future use.

National Book Foundation

1.7 Software Engineering and Hardware Engineering

Software Engineering: It is a systematic approach to the development, operation, and maintenance of software. It involves applying engineering principles to software development, encompassing techniques, methodologies, tools, and processes to ensure the quality, reliability, and efficiency of software systems. Some common types of Software Engineering are:

Fig.1.44 Software development

- **Application Software Engineering:** This focuses on developing software applications that fulfill specific user requirements, such as web applications, mobile apps, desktop software, etc.
- **System Software Engineering:** This involves designing and developing software that provides a platform for other software to run on, such as operating systems, compilers, device drivers, etc.
- **Embedded Software Engineering:** This refers to the specialized field of software engineering that focuses on developing and testing, software that is embedded within hardware devices or systems. For examples Automotive Embedded Systems, Digital home appliances, Industrial Control Systems, etc.
- **Enterprise Software Engineering:** This involves developing software solutions for large-scale enterprises to automate processes, manage data, and facilitate communication within the organization.
- **Game Development:** This area involves creating video games, including game engines, graphics, audio, and animation.

Hardware Engineering: It involves designing, developing, and testing physical components of computer systems and electronic devices. It focuses on the design and construction of hardware components such as processors, memory devices, circuit boards, sensors, etc. Some common types of Hardware Engineering are:

Fig.1.45 Hardware construction

- Digital Hardware Engineering: It deals with designing and developing digital circuits and components such as processors, memory units, etc.
- Integrated Circuit (IC) Design: It focuses on designing integrated circuits, including CPUs (Central Processing Units), GPUs (Graphics Processing Units), etc.
- Computer Architecture: It deals with designing the structure and organization of computer systems, including CPU architecture and memory hierarchy, etc.
- Embedded Systems Design: It focuses on designing hardware systems that are integrated into larger systems or devices, such as microcontrollers, sensors, etc.

National Book Foundation

1.8 Computer Software

Computer software, often referred to simply as "software," is a collection of programs, data, and instructions that tell a computer how to perform specific tasks or functions. It is an important component of any computer system, enabling it to process data, run applications, and interact with users. Software is typically categorized into two main types: system software and application software.

Computer software can be classified into the following types.

- System Software
- Application Software

1.8.1 System Software

System software refers to a type of computer program that manages and controls the hardware components of a computer system, as well as provides a platform for running application software. It plays a crucial role in enabling the interaction between the user, application software, and the underlying hardware.

System software serves as an intermediary between the user and the hardware, making it easier for users to interact with and utilize computer systems effectively.

The following are some common types of system software.

Operating System (OS):

The operating system is a fundamental type of system software that manages hardware resources and provides services for computer programs. It controls tasks such as process scheduling, memory management, file system management, and hardware device communication. Common examples of operating systems include Microsoft Windows, macOS, Linux, and Android. Some common functions of OS include:

- The OS facilitates user interaction by providing a user-friendly interface.
- It manages input/output operations.
- It looks after the allocation of tasks to the processor.
- It handles the allocation and deallocation of memory to programs.
- It helps in organizing files and directories, as well as provides mechanisms for storage and retrieval.
- It manages peripheral devices, such as printers and storage devices, and provides necessary device drivers.
- It provides security and access control through user authentications like User Identifications, passwords and PINs etc.

Device Drivers:

Device drivers are software components that facilitate communication between the operating system and hardware devices like printers, graphics cards, and network adapters. They ensure that the OS can interact with these devices correctly.

Utilities:

System utilities are tools that help manage and maintain the computer system. They

National Book Foundation

can perform tasks such as disk cleanup, data backup, system monitoring, and virus scanning. Examples include disk defragmenters, antivirus software, and system diagnostic tools.

Compiler and Assembler:

These tools are essential for converting high-level programming languages (like C++, Java, or Python) into machine code that the computer's processor can understand. Compilers translate high level language code (source code) into executable programs, while assemblers do a similar job for assembly language code. Fig. 1.46 shows the compilation process.

Fig. 1.46 Compilation process

Linkers and Loaders:

Linkers and loaders are programs that help with the execution of programs. Linkers combine multiple object files (compiled code) into a single executable file, while loaders load these files into memory for execution.

Firmware:

Firmware is a type of software that is permanently stored on hardware devices. It provides low-level control over the device's operation. Examples include the BIOS (Basic Input/Output System) in a computer's motherboard or the firmware in a digital washing machine.

1.8.2 Application Software

Application software, often referred to as "apps" or "software applications," is a category of computer programs designed to perform specific tasks or functions for computer users. Unlike system software, which manages and controls the hardware and provides a platform for running applications, application software is created to address the various needs and requirements of users.

Some examples of application software are:

- Productivity Software
- Business Software
- Entertainment Software
- Educational Software

Productivity Software

Productivity software is designed to help users perform tasks efficiently, organize information, and create contents like documents, presentations, spreadsheets and databases. It includes software that facilitate office work, document management, and collaboration.

National Book Foundation

Examples:

- Microsoft Office Suite: Includes applications like Microsoft Word (word processing), Excel (spreadsheets), and PowerPoint (presentation).
- Google Workspace: Offers tools like Google Docs, Sheets, and Slides for online collaboration and document creation.
- LibreOffice: A free and open-source office suite with applications similar to Microsoft Office.

Business Software

Business software are specifically designed to meet the needs of businesses and organizations. These software aim to streamline and enhance various aspects of business operations, ultimately improving efficiency, productivity, and decision-making.

Examples:

- QuickBooks: Accounting software for managing financial transactions and generating reports.
- Salesforce: Customer Relationship Management (CRM) software for sales and marketing.
- Trello: Project management tool that helps teams organize tasks and collaborate on projects.

Entertainment Software

Entertainment software is designed for leisure and enjoyment. It includes a wide range of applications, from video games to multimedia players and streaming services.

Examples:

- Minecraft: A popular game that allows players to build and explore virtual worlds.
- Spotify: A music application that offers a vast library of songs and playlists.
- Netflix: An online streaming service for movies, TV shows, and documentaries.

Educational Software

Educational software is created to support learning and skill development. It includes a variety of applications and tools that support educational activities, ranging from interactive learning games to digital resources for teaching and assessment.

Examples:

- Learning Management Systems (LMS): LMS platforms provide a centralized place for educational content, resources, assessments, and communication between educators and students. Some examples are ULearn, Virtual Academy, FBISE LMS, etc.
- Kahoot!: An online learning platform that allows educators to create interactive quizzes and games for students.
- Duolingo: Language learning app that gamifies the process of learning new languages.

National Book Foundation

Skills Development

To meet curriculum standards, students will be required to engage in hands-on activities, practical exercises, and real-world projects to develop proficiency in using digital tools such as Word Processors, Spreadsheets, Presentation software, and graphics software for effective communication. Emphasis will be placed on fostering critical thinking, creativity, and problem-solving skills, empowering students to confidently express themselves across various mediums in the digital realm.

Introduction to Common tools:

1. Image Processing Tools:

- Adobe Photoshop: A professional-grade software used for editing and manipulating images. It offers a wide range of features including photo retouching, graphic design, and digital painting.
- Canva.com: A user-friendly online graphic design platform suitable for creating various visual content such as social media graphics, presentations, posters, and more. It provides pre-designed templates and easy-to-use editing tools.
- GIMP (GNU Image Manipulation Program): An open-source alternative to Photoshop, offering many of the same features for image editing and manipulation. It's free to use and has a strong community of users and developers contributing to its development.

2. Word Processors:

- Microsoft Word: A widely-used word processing software offering features for creating, editing, and formatting text documents. It includes tools for spell-checking, grammar checking, and collaboration.
- Google Docs: An online word processor that allows real-time collaboration on documents. It offers similar features to Microsoft Word but is accessible through a web browser and allows multiple users to work on the same document simultaneously.

3. Presentations:

- Microsoft PowerPoint: A presentation software used to create slideshows for professional or educational purposes. It provides tools for designing slides, adding multimedia elements, and delivering presentations effectively.
- Google Slides: A web-based presentation program offered by Google as part of its office suite. It allows users to create, edit, and collaborate on presentations online. Similar to Google Docs, it enables real-time collaboration and sharing.

4. Spreadsheets:

- Microsoft Excel: A powerful spreadsheet program used for organizing, analyzing, and visualizing data. It offers a wide range of functions, formulas, and charting tools to manipulate and present data effectively.
- Google Sheets: An online spreadsheet application that allows users to create, edit, and collaborate on spreadsheets in real-time. It offers similar functionalities to Microsoft Excel but is accessible through a web browser and allows for easy sharing and collaboration.

National Book Foundation

Instructions for Schools/Instructors on utilizing various Tools for Student Learning:

By following these instructions, Schools and IT instructors can effectively introduce students to a variety of tools, provide hands-on practice opportunities, assign meaningful tasks and projects, and facilitate a conducive learning environment for skill development and academic success.

1. **Introduction to Tools:** Begin by introducing each tool to students, providing an overview of its purpose, features, and relevance to their learning objectives.
2. **Access and Installation:**
   - For software tools like Photoshop and Microsoft Office suite, ensure that they are installed on school computers or provide instructions for students to install them on their personal devices.
   - For online tools like Canva.com, Google Docs, Google Slides, and Google Sheets, ensure students have access to the Internet and guide them through the process of accessing these tools through web browsers.
3. **Hands-on Practice Sessions:**
   - Schedule dedicated hands-on practice sessions where students can familiarize themselves with each tool.
   - Provide step-by-step tutorials or demonstrations on how to navigate through the interface, use basic features, and execute common tasks.
   - Encourage students to explore the tools independently, experimenting with different functions and tools to enhance their proficiency.
4. **Assign Practical Tasks and Projects:**
   - Assign practical tasks aligned with the curriculum to reinforce learning and allow students to apply their newfound skills.
   - For example, in image processing tools like Photoshop or Canva.com, assign tasks such as designing posters, editing photographs, or creating digital artwork.
   - In word processors like Microsoft Word or Google Docs, assign writing assignments, essay projects, or collaborative document creation tasks.
   - Utilize presentation tools like Microsoft PowerPoint or Google Slides for student presentations, class reports, or multimedia projects.
   - Incorporate spreadsheets such as Microsoft Excel or Google Sheets for data analysis exercises, budget planning tasks, or scientific experiments.
5. **Provide Guidance and Support:**
   - Offer guidance and support to students as they work on their tasks and projects, addressing any questions or challenges they may encounter.
   - Encourage peer collaboration and discussion to foster a collaborative learning environment where students can learn from each other's experiences and

National Book Foundation

- Scratch: A visual programming language for teaching coding concepts to children.

1.8.3 Programming Languages

A programming language is a structured and systematic method of communicating instructions to a computer. It consists of a set of predefined commands, syntax, and rules that allow programmers to write instructions, enabling the computer to perform specific tasks or solve problems. It serves as a means of communication between a human programmer and a computer, facilitating the development of software and applications.

Programming languages can be classified into two categories, that is, low level languages and high level languages.

Low Level Languages

Low level language is machine-oriented language. To understand low level language, detailed knowledge of internal working of computer is required. Low level languages include machine language and assembly language.

Machine Language: Programming language that is directly understood by computer hardware is known as machine language. Machine language is associated with architecture of computer. Therefore, programs written in machine language for one computer will not work on another because of design differences. It consists of 0s and 1s. It is almost impossible for humans to use machine language because it entirely consists of numbers. Therefore, practically no programming is done in machine language. Instead, assembly languages and high level languages are used.

Assembly Language: Assembly language consists of symbolic codes or abbreviations known as **mnemonics**. It was developed to make computer programming easier than machine language. The abbreviations used in assembly language make it easier to learn and write programs compared to machine language. A program written in assembly language must be converted into machine language before it is executed by computer. A program known as assembler is used to translate assembly language into machine language. Some important characteristics of Assembly language are:

- Assembly language allows programmers to have access to all the special features of the computer they are using. Certain types of operations which are not possible in high level languages are easily programmed using assembly language.
- Generally a program written in assembly language will require less storage and less running time than one prepared in a high level language.
- Assembly languages are still the best choice in some applications but their use is gradually declining.

High Level Languages (HLLs)

High level languages are English-oriented languages and they are commonly used for writing computer programs. These languages use English language words such as **print, goto, if, end,** etc. Therefore, they are easy to learn and use. Some examples of high level languages are **Visual Basic, C, Java** and **Pascal.**

A program known as compiler/interpreter is required to translate a high level program into machine language. Coding and debugging of a high level language program is

National Book Foundation

much easier than a program written in a low level language.

High-level languages can be classified into procedural, structured and object-oriented programming languages.

Procedural and Structured Languages

Procedural programming is based upon the concept of modular programming. In modular programming, programs are divided into smaller parts known as modules. Modular programs consist of one or more modules. A module is a group of statements that can be executed more than once in a program. Each module in a program performs a specific task. It is easy to design, modify and debug a program in a procedural language since it provides better programming facilities.

Structured languages consist of three fundamental elements, which are sequence, selection and repetition

Sequence: It means, writing program statements in a logical sequence. Each step in the sequence must logically progress to the next without producing any undesirable effects.

Selection: It allows the selection of any number of statements based on the result of evaluation of a condition which may be true or false. Examples of statements that implement selection in programming are if, else-if, switch, etc.

Repetition (loops): It means executing one or more statements a number of times until a condition is satisfied. Repetition is implemented in programs using statements, such as for and while loops.

Some examples of structured and procedural languages are FORTRAN, Pascal, C, BASIC, ALGOL, PL/1 and Ada Pascal.

Object-Oriented Programming Languages

Object-oriented programming (OOP) refers to a programming method that is based on objects such as student, vehicle, building, etc. Object-oriented programming language provides a set of rules for defining and managing objects. An object can be considered a thing that can perform a set of activities. For example, the object vehicle can be defined as an object that has number of wheels, number of doors, color, number of seats, etc. The set of activities that can be performed on this object include Steer, Accelerate, Brake, etc.

Complicated large computer programs are difficult to design, develop, maintain and debug. The concept of object-oriented programming solves this problem. The most widely used object-oriented programming languages are C++, Visual Basic and Java.

Teacher's Guide

- Utilize examples and demonstrations to illustrate how system software interacts with hardware to enable efficient computer operations.
- Encourage students to explore examples of application software relevant to their academic pursuits to deepen their understanding.
- Provide hands-on coding exercises or interactive demonstrations to familiarize students with basic programming concepts and syntax.

National Book Foundation

Uses of Low Level Languages

Important uses of low-level programming languages include:

Use	Explanation
Operating System Development	Writing the core software that manages hardware resources.
Device Drivers	Creating software to enable communication with hardware devices.
Embedded Systems	Programming microcontrollers and IoT devices for specialized functions.
Firmware Development	Developing software that resides on hardware components.
Real-Time Systems	Ensuring precise timing and responsiveness in industrial control, robotics, and aerospace systems.
Security Tools	Building intrusion detection, firewalls, and encryption software for robust security.
Game Development	Optimizing game engines, physics simulations, and graphics rendering for performance

Uses of High Level Languages

Important uses of high-level programming languages include:

Use	Explanation
Applications (Apps) Development	Creating desktop, mobile, and web applications (Apps) for various platforms.
Web Development	High-level languages are used to build websites, making them interactive and functional.
Data Analysis and Science	Analyzing large datasets and conducting scientific research.
Machine Learning and AI	Developing machine learning models and AI algorithms.
Automation and Scripting	Automating tasks and processes, including system administration and data manipulation.
Game Development	Developing gameplay logic, AI, and user interfaces for games.
Database Management	Creating, querying, and managing databases.
Scientific and Engineering Simulations	Simulating complex systems and conducting simulations.
Business Software	Developing enterprise-level software for various industries.
Educational Tools	Creating e-learning platforms and educational software.

National Book Foundation

1.9 Data Communication

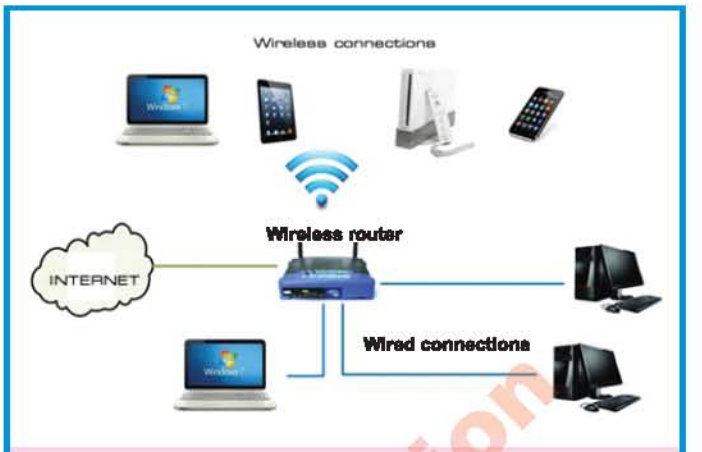


Fig.1.47 A Communication Network

Data communication refers to the process of exchanging data or information (through a computer network) between two or more devices or systems through a transmission medium such as cables, optical fibers, or wireless mediums. This communication can involve the transfer of various types of data, including text, numbers, images, audio, and video, and it is a fundamental component of modern information technology and telecommunications.

A simple communication network having wired and wireless connections is shown in Fig.1.47.

1.9.1 Network Communication Components

Data communication is the process of transferring information from one point to another in a networking environment. Network communication consists of five basic components, as shown is Fig.1.48.

- Sender
- Message
- Medium
- Protocol
- Receiver

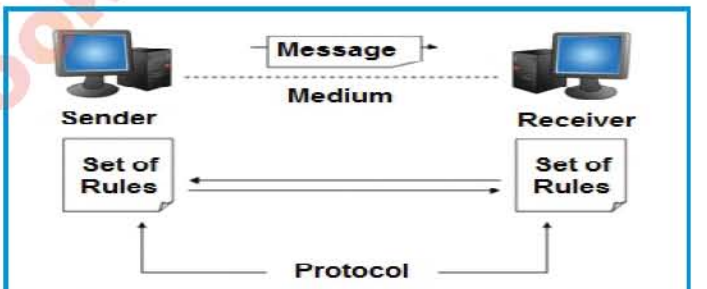


Fig.1.48 Components of Network Communication

Sender

Sender, also called transmitter is a computer/device that sends the message (data or information) from source to destination in a communication network. It may be a computer, workstation, cell phone or camera. The sender device converts the electrical signal into a form that is suitable for transmission over the communication network.

Message

Message is the data or information that is to be transmitted. Message can be in the form of text, audio, video, or any combination of these.

Medium

Medium is the path through which message travels from source to destination.

National Book Foundation

Medium can be wired, for example telephone cable, coaxial cable and fibre optics. It can also be wireless for example Bluetooth, Wi-Fi, microwave, radio wave and satellite.

Receiver

Receiver is the device which receives transmitted message. It can be a computer, workstation, telephone handset or television set. The data received from the transmission medium may not be in proper form to be accepted to the receiver and it must be converted to appropriate form before it is received.

Protocol

A protocol is a set of rules that governs data communications. It represents an agreement between the communicating devices. Without a protocol, two devices are connected but may not communicating with each other.

1.9.2 Modes of Network Communication

Modes of network communication refer to the methods or the ways information is transmitted from one place to another.

The following are different modes of data communication

- Simplex, Half-duplex and Full-duplex
- Synchronous and Asynchronous

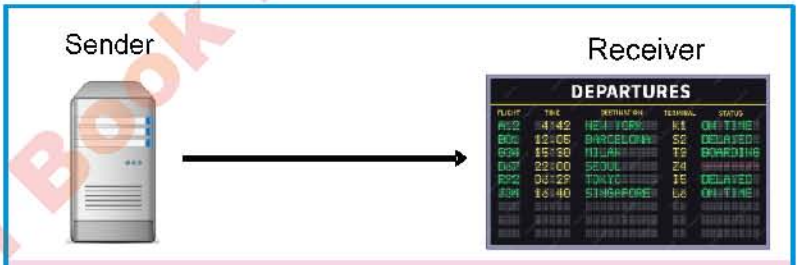


Fig.1.49 Simplex mode of data Communication

Simplex mode

In Simplex mode, the communication takes place in only one direction. In this mode communication is unidirectional, i.e. the communication can only take place in one direction and it is not possible for the receiver to send data back. For example data being sent to an electronic notice board found in train stations and Airports as shown in Fig 1.49. Radio and television broadcastings are also examples of simplex transmission.

Half-duplex mode

In half-duplex mode, the communication takes place in both the directions but not at the same time. The signal can only be sent or received at one time. A common example of this type of communication is the use of walkie-talkies, as shown in Fig. 1.50 where each of the persons communicating must

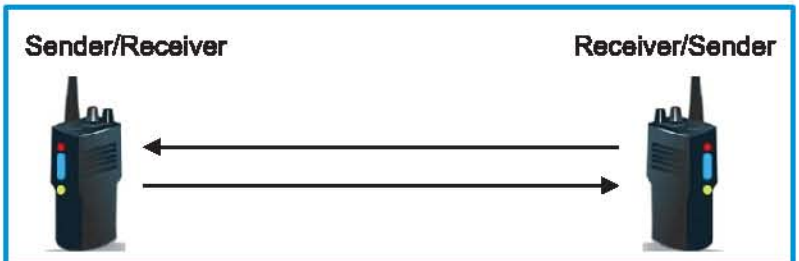


Fig.1.50 Half-duplex mode of data communication

National Book Foundation

indicate when they have finished speaking. Half-duplex transmission is used also in transaction-oriented systems, for example communication between a computer and credit card machine.

Full-duplex mode

In full-duplex mode, the communication takes place in both the directions at the same time. In this mode, both sender and receiver can send and receive the data simultaneously. For example when two computers communicate with each other to send and receive some data, as shown in Fig. 1.51. It is the fastest bi-directional mode of communication. The full-duplex mode is like a two way street, with traffic flowing in both directions at the same time.

Fig.1.51 Full-duplex mode of data communication

One common example of full-duplex communication is the telephone network. When two people are communicating by a telephone line, both can talk and listen at the same time.

Asynchronous Transmission

In asynchronous transmission, the time interval between two characters is variable and not fixed as shown in Fig.1.52. The computer devices can exchange information at their own rate, slow or fast. Start and Stop bits are used in asynchronous transmission. These bits provide timing (synchronization) for the connection between the sender and the receiver. The start bit tells the receiver that a character is coming and stop bit indicates that the transmission of character has ended. This type of transmission is ideal for slow-speed communication when gaps may occur during transmission. Example of asynchronous transmission is keyboard data transmission.

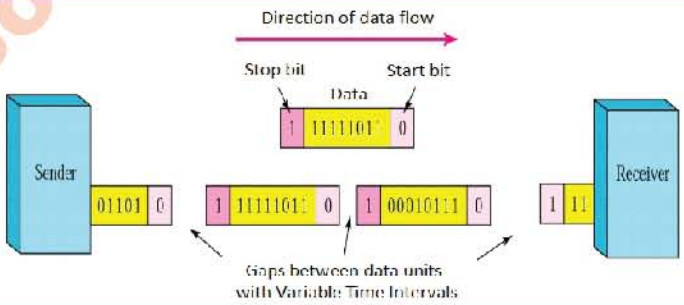


Fig.1.52 Asynchronous transmission

Synchronous Transmission

In synchronous transmission, the time interval between two characters is always the same as shown in Fig.1.53. In this method two communicating devices are synchronized and they continue to send characters in order to remain synchronized, even if there is no data to be transmitted. A special "idle" character is sent when there is no data for transmission. It does not require transmission of start and stop bits. It sends data as one long bit stream or

National Book Foundation

block of data and each bit is sent one after the other. The receiver counts the bits and reconstructs the sent information in bytes. It is essential that timing is maintained as there are no start and stop bits and no gaps. Accuracy is dependent on the receiver keeping an accurate count of the bits as they come in.

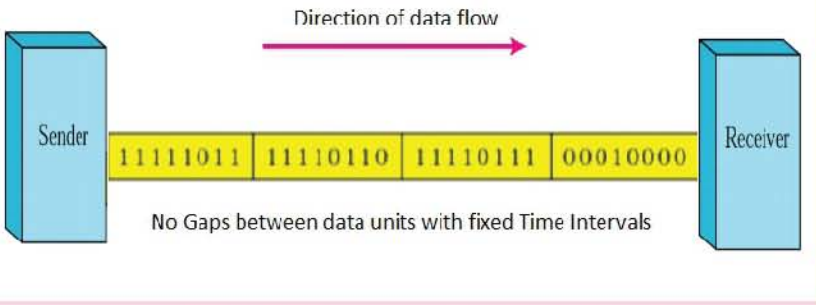


Fig.1.53 Synchronous transmission

Synchronous transmission is faster than asynchronous because fewer bits have to be transmitted; i.e. only data bits and no extra control bits are sent. The best example of synchronous transmission is the data transmission between devices in network communications links.

1.9.3 Communication Devices

A device that is used in telecommunication systems for transmitting data from one location to another is known as communication device.

Commonly used communication devices are: Hub, Switch, Router and Gateway.

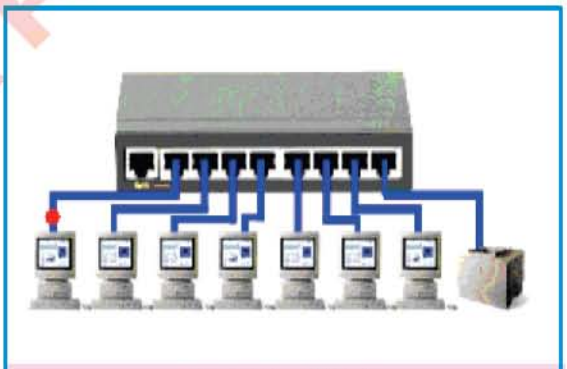


Fig.1.54 a 8 Hub

Hub

Hub is a connectivity device used in LAN. It connects multiple LAN devices on one network and makes them act together as a single network. A hub is non-intelligent device and sends output to all the devices on the network. A hub has multiple input/output (I/O) ports, in which an input in one port results in it being an output in all the other ports, except the port where it was input. A hub is shown in Fig. 1.54.

Fig.1.55 8 Port Switch

Switch

Switch is a networking device that performs the same job as the hub but are considered as intelligent than hub. It gathers information about the data packet and forwards it to only the node (e.g. computer) it was intended for. A data packet is a basic unit of

National Book Foundation

communication over a computer network. When data is transmitted, it is broken down into packets which are reassembled to the original form once they reach the destination. A switch is shown in Fig. 1.55.

Router

Router is a communication device which is used to connect two or more networks. Today, most of the networks are connected to Internet. When the computer is sending data to another computer on the Internet, router receives the data packets, looks for the remote computer address and forwards it to a computer that is closer to the remote computer. It forwards the data packets by selecting the best path-way based on network traffic. Many routers take part in transmitting the data packets from one location to another. A wireless router is shown in Fig.1.56.

Fig. 1.56 Wireless Router

Gateway

Gateway is a device that is used to connect a network to another network that uses different protocols. If we have to link different kinds of networks, such as a network of IBM mainframe computers and a network of PCs, we might have to use a gateway. Gateways change the format of the data packets but not the contents of the message, to make it conform to the application program of the remote computer. A gateway is shown in Fig. 1.57.

Fig.1.57 Gateway Device

1.9.4 Network Architecture

Network architecture is the design of a communication system. It includes hardware devices (such as routers and switches), cabling, network topology and physical and wireless connections. Computer networks consist of server computers and client computers.

Server Computer: A computer on the network that shares resources for others to use is called a server computer or simply server. Shared resources include information, software, printer, plotter, Internet connection, hard disk, etc.

Client Computer: A computer on the network that accesses resources that are shared by other computers is known as client computer or simply client.

National Book Foundation

The two commonly used network architectures are:

- Client/Server Network
- Peer-to-Peer Network

Client/Server Networks

A computer network in which each computer on the network acts as either a server or a client is called client/server or dedicated server network. Each server computer on the network is called a dedicated server. Servers are not used as client computers. Fig.1.58 illustrates how a dedicated server network may be designed. The computer at the top of the figure is the dedicated server, sharing files and applications. The remaining computers in the illustration are clients that access resources shared by the server. Similarly, in a dedicated server network, client computers never act as servers.

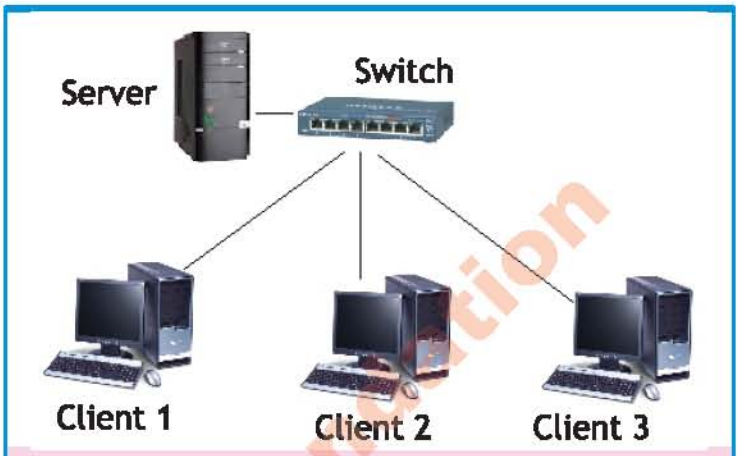


Fig.1.58 Client/Server Network

Client/Server network includes one or more computers that are dedicated to acting as servers. The servers are optimized to provide quick access to shared network resources. Servers also provide centralized security to ensure that resources are not accessed by unauthorized users.

Because the client/server approach centralizes control of data and other shared resources, one person or group is typically responsible for administering the network.

Peer-to-Peer Networks

In Peer-to-Peer networks, every computer is capable of playing the role of client, server or both at the same time. In this network each computer on the network is referred to as peer. In a peer-to-peer network, a peer computer can act as both a server and a client at the same time. A peer computer on your desktop can share files and printers with other computers and it can simultaneously access other shared resources on the network. A conceptual view of a peer-to-peer network is shown in Fig.1.59.

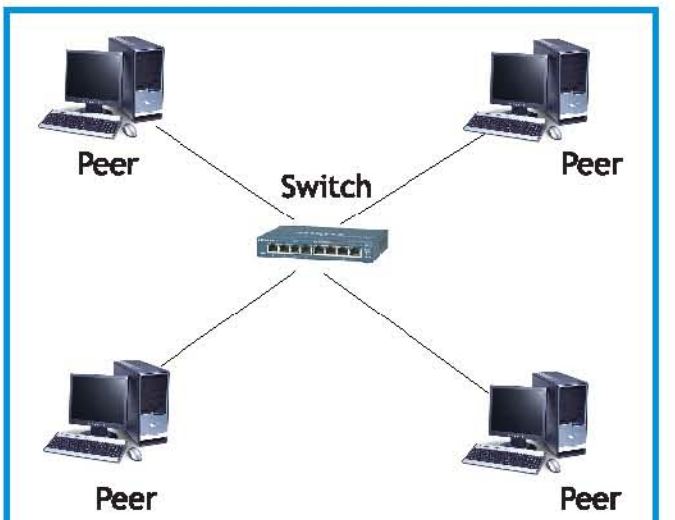


Fig.1.59 Peer-to-Peer Network

National Book Foundation

Peer-to-peer networks tend to be relatively small. Most of these networks fall to range between two and ten computers. Large peer-to-peer networks become difficult to manage, because so many network administrators control sharing and maintaining shared resources.

1.9.5 Types of Networks

The following are different types of networks based on the size and physical area they cover.

Local Area Networks

A Local Area Network (LAN) spans a limited physical area. It is confined to a single building or a group of nearby buildings. LANs are used for sharing applications, printers, group scheduling, e-mail, project tracking and other tasks. A LAN is shown in Fig.1.60.

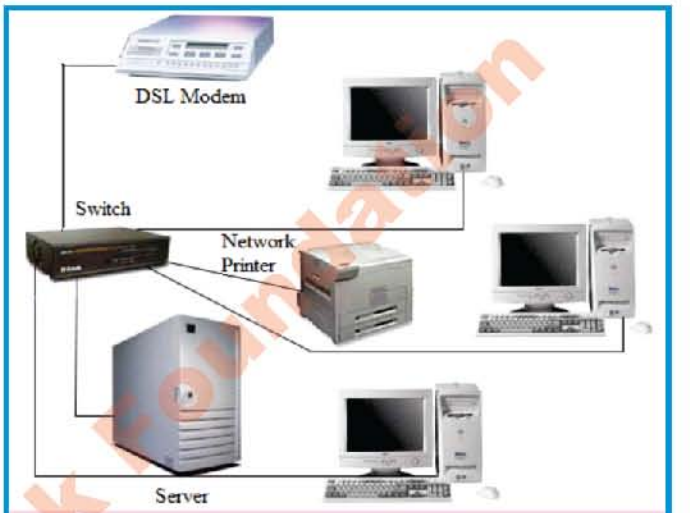


Fig.1.60 Local Area Network

Characteristics of LAN

- Spans a small physical area.
- Uses high-speed wired/wireless connections between computers.
- It is a very reliable network. Communication errors are very rare.
- It consists of a limited number of computers.

Wide Area Networks

A Wide Area Network (WAN) spans a large physical area, connecting several sites of an organization across cities, countries and continents. Because of the longer distances involved, WANs are sometimes referred to as long-haul networks.

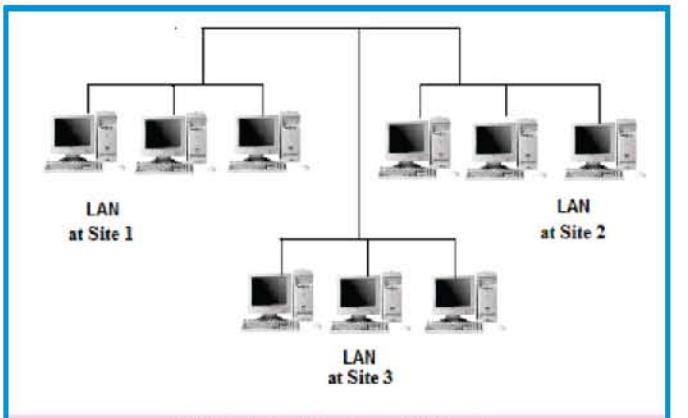


Fig.1.61 A Wide Area Network

A WAN is often made up of two or more LANs connected together as shown in Fig.1.61. For example, you might have a LAN at each site of your organization and each of those LANs might be connected together to form a WAN.

Characteristics of WAN

- Spans a large physical area. It can be worldwide like Internet.
- Communication speed is slow compared to LAN.

National Book Foundation

- Connects computers through public networks, leased lines or satellites.
- Connects multiple LANs.
- Sometimes communication errors occur due to its complexity.

Metropolitan Area Network

Metropolitan Area Network (MAN) can span from several buildings or a large campus to entire cities. MAN is used by many organizations. It also connects a number of local area networks with high-speed communication lines.

Characteristics of MAN

- It is larger than a LAN and smaller than a WAN. Covers an area of between 5 to 50 km diameter.
- Uses fiber optic cable or microwave transmission.
- Provides high-speed communication.
- Used by telephone companies, Internet Service Providers and cable TV companies.

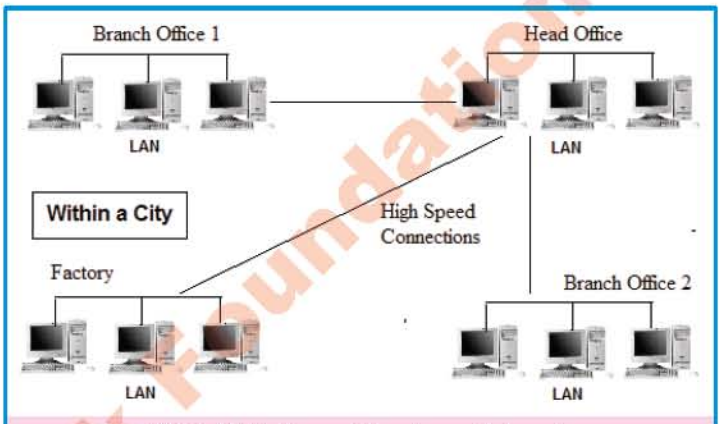


Fig.1.62 Metropolitan Area Network

Virtual Private Network

Virtual Private Network (VPN) is a computer network that provides remote access to individuals and offices to their organization's networks. It provides cheap communication by using public telecommunication infrastructure such as Internet instead of expensive leased lines. It allows employees at home or on trip to connect their laptops into the computer as office through public telecommunication networks and do their work.

Characteristics of VPN

- It uses public networks such as Internet to connect computers.
- Provides secure remote access.
- Enables files sharing, video conferencing and similar network services.
- Provides cheap communication over long distance.

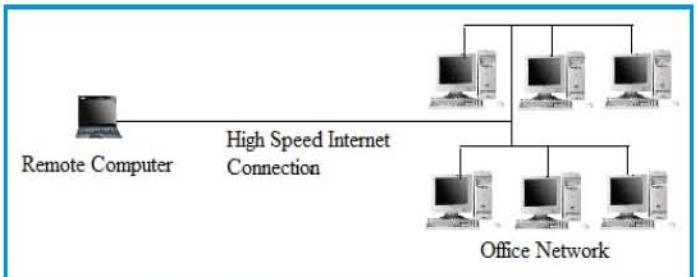


Fig.1.63 Virtual Private Network

Teacher's Guide

- Engage students in hands-on activities or simulations to simulate network configurations and troubleshoot common network issues.

National Book Foundation

1.9.6 Wireless Networks

Wireless networks refer to networks that use wireless communication technologies to transmit data between devices without the need for physical wired connections. Wireless networks can operate over short distances (e.g., within a building) or cover large geographical areas (e.g., cellular networks). Here are some examples and uses of wireless networks:

Fig.1.64 Wireless headset

Examples of Wireless Networks:

- **Wi-Fi (Wireless Fidelity):** Wi-Fi is a common wireless networking technology that enables devices to connect to local area networks (LANs) and access the internet wirelessly. It is widely used in homes, offices, airports, and public places.
- **Bluetooth:** It is a short-range wireless technology used for connecting devices in close proximity. It is commonly used for wireless audio streaming, file sharing, connecting peripherals (e.g., keyboards, mouse), etc. Fig. 1.64 shows a wireless headset for voice streaming.
- **Cellular Networks:** They provide wireless communication over large geographic areas, divided into cells served by base stations (cell towers). They enable mobile voice calls, messaging, internet access, and other data services to mobile devices.
- **Satellite Networks:** They use communication satellites orbiting the Earth to provide wireless connectivity over large areas. They are used for satellite Internet access, satellite TV broadcasting, GPS (Global Positioning System) navigation, etc.

Advantages of Wireless Networks:

- Wireless networks enable users to access the internet and network resources from anywhere within the coverage area, offering mobility and convenience.
- They are easier and more cost-effective to install.
- Wireless networks can easily expand to accommodate additional devices and users.
- Users benefit from the ability to connect multiple devices to the same network for communication and data sharing.
- Wireless networks save costs by eliminating the need for expensive cabling.
- Wireless networks facilitate remote access to network resources, enabling individuals to work from home or access corporate resources while on the go.

Disadvantages of Wireless Networks:

- Wireless networks can be affected by interference from other devices, neighboring networks, and environmental factors, which can cause performance problems.
- Security risks, like unauthorized access and data breaches.
- Wireless networks have a limited range, so users might need extra boosters.
- Sometimes wireless networks can be slower, especially in busy places.
- There can be issues with reliability, like dropped signals or slow connections.
- Wireless devices use power, so they can use more energy than wired devices.

National Book Foundation

1.9.7 Network Topologies

The arrangement of network nodes (any devices which are part of network) and connections between them is called the network's topology. A node represents any device on the network. Topology is simply a map of the layout of nodes and connections in the network. Four network topologies are popular today, namely, Bus, Star, Ring, and Mesh.

Bus Topology

Bus network topology connects each node to the network along a single piece of cable, called a bus. Bus network topology is shown in Fig.1.65.

Features of Bus Topology

- Suitable for a small network.
- Easy to connect a computer or a peripheral device to the network.
- Requires less cable to implement.
- Terminator is installed at each end of the cable to prevent signals from reflecting back onto the bus and cause errors. Terminator is a device that is attached to ground.

Fig.1.65 Bus network topology

Limitations of Bus Topology

- If the single cable is damaged or broken at any point, the entire network can go down.
- Difficult to identify the problem if the entire network goes down.
- Not suitable for large network.

Star Topology

In a star network topology, each network node is connected to a central device called a hub. Large networks can require many hubs and hubs can be connected to each other to create a single large network. Star network topology is shown in Fig.1.66.

Features of Star Topology

- It is suitable for both small and large networks.
- Easy to install and wire.
- Easy to detect and remove faults.

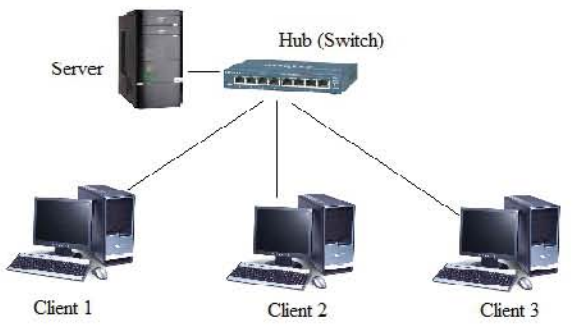


Fig. 1.66 Star network topology

National Book Foundation

- Failure of cable does not stop functioning of the entire network.

Limitations of Star Topology

- Failure of the hub causes the entire network to go down.
- Expensive topology to implement. Lengthy cable with a hub is required to install star topology

Ring Topology

Ring topology is shaped just like a ring. It is made up of an unbroken circle of network nodes. Ring network topology is shown in Fig.1.67.

Features of Ring Topology

- Each node is directly connected to the ring.
- Easy to install and wire.
- Data on the network flows in one direction.
- Not costly to implement.

Limitations of Ring Topology

- If the ring is broken at any point, the entire network stops functioning.
- Slower than other network topologies.

Fig.1.67 Ring network topology

Mesh Topology

In mesh topology, each node is directly connected to all the nodes as shown in Fig.1.68.

Features of Mesh Topology

- Most reliable network topology.
- Data can be routed around failed computers or busy ones.
- Can manage high traffic.

Limitations of Mesh Topology

- Most expensive topology to implement.
- Setup and maintenance is very difficult.

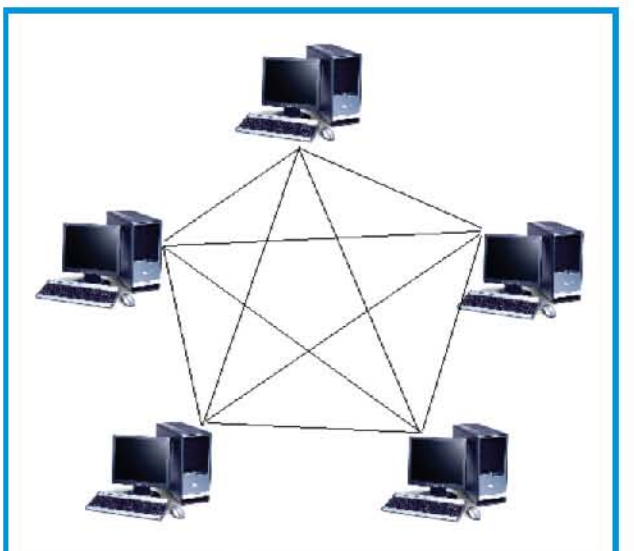


Fig.1.68 Mesh network topology

National Book Foundation

1.9.8 Packet Switching and Circuit Switching

Packet switching and circuit switching are two different approaches to network communication, each with its own advantages and applications. Packet switching is widely used in modern networks, especially for data transmission over the internet, while circuit switching is still used for applications requiring guaranteed bandwidth and predictable performance, such as traditional telephone system and dedicated data connections.

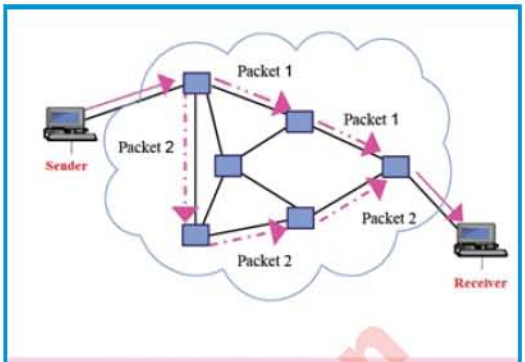


Fig.1.69 Packet Switching

Packet Switching: Packet switching is a networking technology that breaks data into small packets for transmission over a network. Each packet contains a portion of the data, along with source and destination addresses, allowing them to be independently routed across the network to their destination. Packets may take different paths through the network and may arrive out of order, but they are reassembled at the destination. Fig. 1.69 shows the mechanism of packet switching network.

Example: Consider sending an email from one computer to another over the internet using packet switching:

- The email message is divided into smaller packets, each containing a portion of the message, along with source and destination addresses.
- These packets are then transmitted over the internet, possibly taking different routes and traveling through various network devices such as routers and switches.
- At the destination, the packets are reassembled in the correct order based on their sequence numbers and then presented to the recipient as the original email message.

Circuit Switching: Circuit switching is a networking technology that establishes a dedicated communication path (circuit) between two devices for the duration of a communication session. Once the circuit is established, data is transmitted over the path without interruption until the session is terminated. Fig. 1.70 shows the mechanism of circuit switching network.

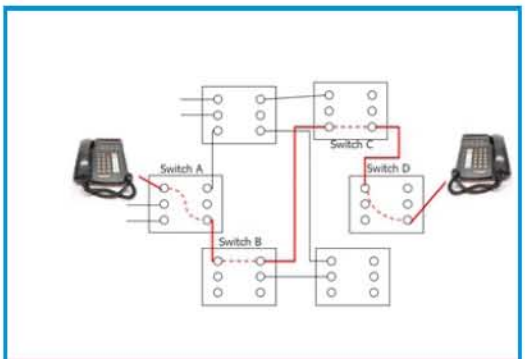


Fig.1.70 Circuit Switching

Example: Making a traditional phone call using circuit switching:

- A dedicated circuit is established between the caller and the recipient when the call is initiated.
- The caller and recipient can communicate with each other using the allocated circuit until one of them hangs up, at which point the circuit is released.

National Book Foundation

1.9.9 Data Communication Standards

Data communication standards are a set of rules (protocols and specifications) that define how data is transmitted, received, and processed in computer networks and communication systems. These standards ensure that devices and systems from different manufacturers can communicate and work together flawlessly. Data communication standards play a crucial role in enabling interoperability, reliability, and efficiency in the exchange of data.

The OSI (Open Systems Interconnection) is one such standard conceptual framework used in the field of computer networking to define and understand how different networking protocols and technologies interact and work together.

1.9.10 OSI Model

The International Standards Organization (ISO) based in Geneva, developed standards for international and national data communications. In the early 1970s, ISO developed a standard model of a data communication system and called it the Open Systems Interconnection (OSI) model.

The OSI model consists of seven layers. Each layer performs a specific task during data communication as shown in Fig.1.71.

The seven layers of OSI model are described below.

Layer 7 - Application Layer

Application Layer provides services to end-user. It interacts with the operating system or application software whenever the user wants to send files, read messages or perform other network related activities.

Layer 6 - Presentation Layer

Presentation Layer takes the data provided by the Application Layer and converts it into a standard format that the other layers can understand. At the receiving end it also formats the information so that it looks the way the user can understand.

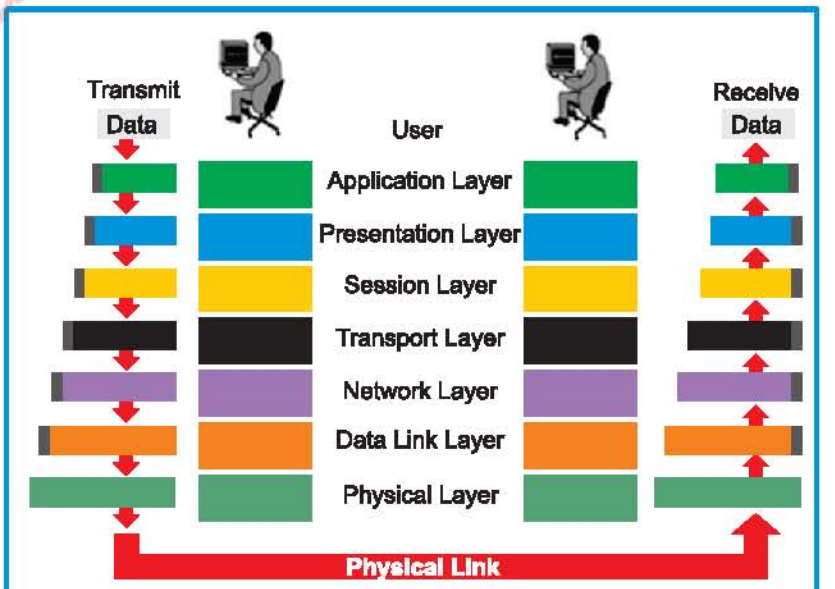


Fig.1.71 Layers of OSI Model

Layer 5 - Session Layer

Session Layer performs functions that enable two applications or two pieces of the same application to

National Book Foundation

communicate across the network. It performs security, name recognition, logging and other similar functions. It also establishes, maintains and ends communication with the receiving computer.

Layer 4 - Transport Layer

Transport Layer establishes connections between two computers on the network. It handles quality control by making sure that the data received is in the right format and the right order.

Layer 3 - Network Layer

Network Layer decides which physical path-way the data should take to reach the destination. The communication device Router works in network layer.

Layer 2 - Data Link Layer

Data Link Layer defines the format of data on the network. This layer converts the data into packets and checks them before putting them on the path-way. The communication device Switch works in this layer.

Layer 1 - Physical Layer

Physical Layer defines cables and signalling. It provides hardware means such as cables and connectors for sending and receiving data. Cables, hubs and repeaters work in this layer.

1.9.11 Data Communication Protocols

Different communication protocols define how data is transmitted and received over a network. Examples include:

- TCP/IP (Transmission Control Protocol/Internet Protocol): Used for internet communication and provides reliable, connection-oriented data transfer.
- HTTP (Hypertext Transfer Protocol): Used for transferring web pages and related data on the World Wide Web.
- FTP (File Transfer Protocol): Used for transferring files between computers on a network.
- SMTP (Simple Mail Transfer Protocol): Used for sending email messages.

1.9.12 The Internet

The Internet is a global network of interconnected computer networks that allows for the exchange of data, information, and communication among users and devices across the world. It is a vast and decentralized network that spans continents and connects billions of computers, servers, and other devices.

Evolution of the Internet

The Internet has evolved from its origins in the 1960s as ARPANET, a U.S. Department of Defense project for research institutions and military installations, to become a global

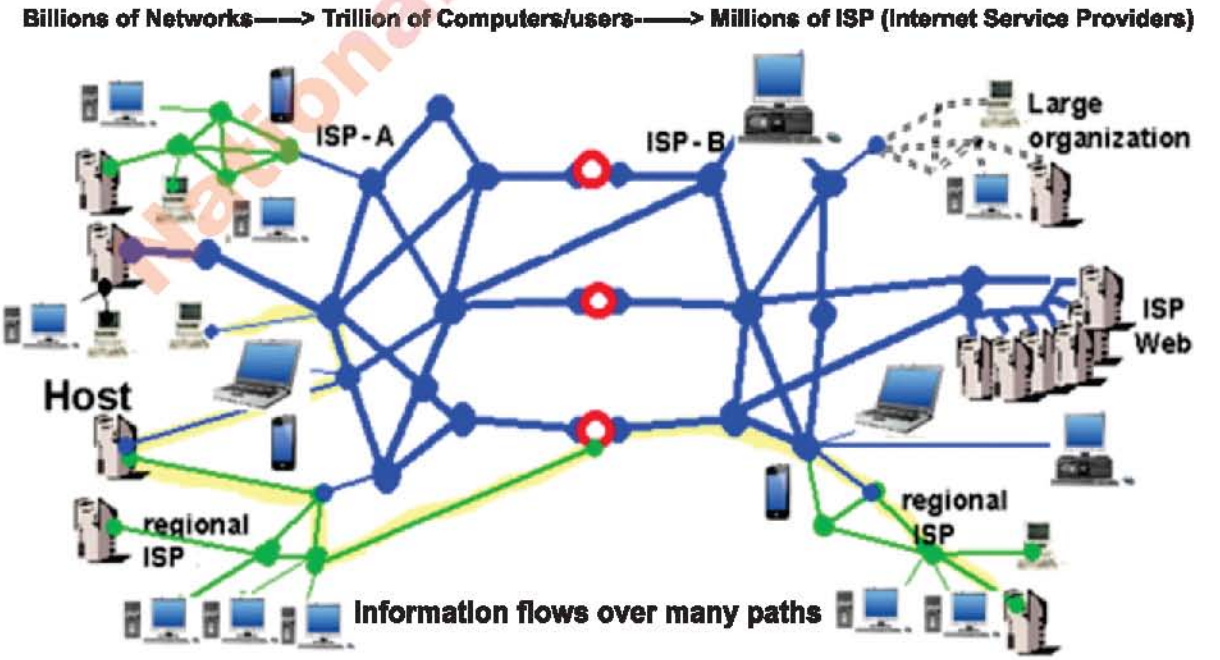
National Book Foundation

network of interconnected computer networks. In the 1970s, the development of TCP/IP protocols established the foundation for the modern Internet, allowing different networks to communicate. The 1990s saw the emergence of the World Wide Web and web browsers, revolutionizing how people access and share information over the Internet. The 2000s brought broadband internet and social media platforms, while the 2010s saw the rise of mobile internet and the Internet of Things (IoT). In the 2020s, the Internet continues to evolve with cloud computing, artificial intelligence, and 5G technology, impacting nearly every aspect of modern life.

Working of the Internet

The Internet is the largest computer network ever built. It globally connects billions of devices and networks. It operates through a decentralized architecture using packet-switching technology. Data is divided into packets, which are routed through a network of interconnected routers and switches. Protocols like TCP/IP ensure data is packaged, addressed, and transmitted correctly. The Domain Name System (DNS) translates human-readable domain names (e.g., www.ncc.gov.pk) into IP addresses. Content is hosted on servers, and data is transmitted as packets to and from these servers. As data travels through the network, it is encapsulated in headers at each layer of the OSI model (such as the IP and TCP headers). When the data reaches its destination, these headers are removed through a process called de-capsulation. Security measures like encryption protects data during transmission. The Internet's interoperable design allows diverse devices and networks to communicate, making it a global information and communication platform.

National Book Foundation



Advantages of the Internet

The main advantages of the Internet include:

- **Global Connectivity:** Enables communication and access to information worldwide.
- **Vast Information:** Provides a vast information resources and knowledge.
- **Communication:** Facilitates real-time communication and collaboration.
- **E-commerce:** Allows online shopping and digital transactions.
- **Education:** Supports online learning and research.
- **Business:** Enhances productivity and global reach.
- **Entertainment:** Offers streaming, gaming, and social media.
- **Innovation:** Promotes technological advancements and research.

Disadvantages of the Internet

The main disadvantages of the Internet include:

- **Privacy Concerns:** Threats to personal data and online privacy.
- **Cybersecurity Risks:** Vulnerability to hacking and cyberattacks.
- **Information Overload:** Overwhelming amount of data and misinformation.
- **Digital Addiction:** Excessive screen time and online dependency.
- **Digital Divide:** Unequal access to the Internet worldwide.
- **Online Harassment:** Cyberbullying and harassment issues.
- **Health Concerns:** Physical and mental health impacts.

Common Applications of the Internet

Main applications of the Internet include:

- **Communication:** Email, messaging, and video calls.
- **Information Retrieval:** Web browsing, search engines, and online databases.
- **E-commerce:** Online shopping, banking, and digital payments.
- **Social Media:** Networking, content sharing, and social interaction.
- **Entertainment:** Streaming, online gaming, and multimedia content.
- **Education:** Online courses, research, and e-learning platforms.
- **Business and Work:** Remote work, collaboration, and e-commerce.
- **Research and Innovation:** Access to research materials and innovation platforms.

National Book Foundation

Summary

- **Computer systems:** are an integral part of modern life, transforming work, communication, learning, and entertainment. They consist of intricate hardware and software components that collaborate to process information and accomplish various tasks.
- **Abacus:** The Abacus, invented by the Chinese 4000 years ago, was a wooden frame with metal rods and beads used for simple arithmetic calculations.
- **Napier's Bone:** John Napier developed Napier's bones, which used 9 strips marked with numerals for multiplication and division, introducing the decimal point system.
- **Pascaline:** Blaise Pascal's 1642 invention, the Pascaline, was the first mechanical calculator, consisting of gears and wheels.
- **Stepped Reckoner or Leibniz Wheel:** Wilhelm Leibniz improved Pascal's device in 1673, creating the digital mechanical calculator known as the stepped reckoner using grooved wheels.
- **Difference Engine:** Charles Babbage's early 1820s invention, the Difference Engine, was a steam-powered calculating machine for basic computations.
- **Analytical Engine:** Charles Babbage's Analytical Engine in 1830 was a mechanical computer that used punch cards, capable of solving complex mathematical problems and storing data.
- **Tabulating Machine:** Herman Hollerith's 1890 invention, the Tabulating Machine, based on punch cards, computed statistics and recorded data, leading to the creation of IBM in 1924.
- **Differential Analyzer:** In 1930, Vannevar Bush introduced the first electrical computer, the Differential Analyzer, using vacuum tubes for calculations at a rate of 25 per minute.
- **Mark I:** Howard Aiken's Mark I, a digital computer invented in 1944, could add eight-digit numbers and print results using punched cards; it was 50 feet long and weighed 5 tons.
- **First Generation Computers (1940 - 1956):** First-generation computers used vacuum tubes, were slow, expensive, and huge, relied on punched cards, and had limited memory.
- **Second Generation Computers (1956-1963):** Transistors replaced vacuum tubes in second-generation computers, increasing speed, reliability, and reducing size and cost.
- **Third Generation Computers (1963 - 1971):** Third-generation computers used integrated circuits (ICs) instead of transistors, consuming less power and introducing keyboard and monitor interfaces.

National Book Foundation

- **Fourth Generation Computers (1971 - Present):** Fourth-generation computers introduced microprocessors, becoming faster, smaller, and supporting advanced input/output devices, modern programming languages, and multimedia software.
- **Fifth Generation Computers:** Fifth-generation computers aim to understand natural languages and possess thinking capabilities, relying on Artificial Intelligence (AI) and enabling user commands in any language.
- **System:** A system is a collection of interconnected components working together to achieve specific purposes, found in various aspects of life.
- **Natural Systems:** Natural systems exist in nature, are diverse, and self-regulate.
- **Examples of Natural Systems:** Ecosystems, weather systems, geological systems, hydrological systems, and the solar system.
- **Artificial Systems:** Human-made systems designed to serve specific purposes or solve problems.
- **Examples of Artificial Systems:** Communication systems, information systems, transportation systems, energy systems, manufacturing systems, and healthcare systems.
- **Input Devices:** Used to provide data to the computer, including keyboards, mice, microphones, scanners, barcode readers, digital cameras, and touch screens.
- **System Unit:** The central part of a computer, including the motherboard, microprocessor (CPU), and registers.
- **Memory Types:** Computer memory includes ROM, RAM, and cache memory, each with specific functions and retention properties.
- **Output Devices:** Display text, graphics, and images, including monitors, printers, plotters, and speakers.
- **Data Transmission:** Data moves between components through buses, following an instruction cycle that fetches, decodes, executes, and stores data.
- **Von Neumann Architecture:** Explains how computer hardware and software work together, comprising a CPU, registers, ALU, control unit, and buses.
- **Memory Units:** Memory is measured in bytes, with various units like kilobytes, megabytes, gigabytes, terabytes, petabytes, and exabytes.
- **Memory Built-up:** Memory can be chip, magnetic, or optical, and memory retention power categorizes it as volatile or non-volatile.
- **Chip Memory:** Fast memory used in various devices like RAM, ROM, cache, and SSDs.
- **Magnetic Memory:** Uses magnetized mediums like hard disks and magnetic tapes to store data.

National Book Foundation

- **Optical Memory:** Stores data as pits and lands on optical disks like CDs, DVDs, and Blu-ray disks.
- **Volatile Memory:** Requires power to retain data, examples include RAM, cache, and registers.
- **Non-Volatile Memory:** Retains data even without power, including ROM, flash memory, and storage devices like hard disks and optical disks.
- **Computer Software:** Computer software is a collection of programs, data, and instructions enabling computers to perform tasks.
- **System Software:** Manages and controls hardware, acting as an intermediary between users and hardware. Includes operating systems, device drivers, utilities, compilers, assemblers, linkers, loaders, and firmware.
- **Application Software:** Designed for specific user tasks, such as productivity, business, entertainment, and education. Examples include word processors, spreadsheet software, games, and educational programs.
- **Programming Languages:** Programming languages instruct computers to perform tasks. Low-level languages (machine and assembly) are hardware-oriented, while high-level languages (e.g., C++, Java) are user-friendly. High-level languages can be procedural, structured, or object-oriented.
- **Data Communication:** Involves exchanging data between devices or systems via networks. Key components include sender, message, medium, protocol, and receiver.
- **Modes of Network Communication:** Modes include simplex (one-way), half-duplex (both ways, not simultaneously), and full-duplex (both ways simultaneously). Transmission can be asynchronous (variable timing) or synchronous (fixed timing).
- **Communication Devices:** Common devices include hubs, switches, routers, and gateways. Each serves specific roles in network communication.
- **Network Architecture:** Includes LANs (local), WANs (wide), MANs (metropolitan), and VPNs (virtual private). Defines the physical layout and connectivity of a network.
- **Network Topologies:** Topologies dictate how nodes are connected; options include bus, star, ring, and mesh. Each has unique features and limitations.
- **Data Communication Standards:** Standards (like OSI) establish rules for data transmission, ensuring compatibility and reliability in networks and systems.
- **OSI Model:** The OSI model, developed by ISO, consists of seven layers that perform specific tasks in data communication, including Application, Presentation, Session, Transport, Network, Data Link, and Physical layers.

National Book Foundation

- **Data Communication Protocols:** Various communication protocols, such as TCP/IP, HTTP, FTP, and SMTP, define how data is transmitted and received over networks.
- **Internet:** Internet is a global network of interconnected computer networks that has evolved from ARPANET to the modern era, facilitating communication, information exchange, and connectivity worldwide.
- **Working of the Internet:** The Internet operates through a decentralized architecture, using packet-switching technology, protocols like TCP/IP, DNS for domain translation, and security measures to transmit and protect data.
- **Advantages of the Internet:** The Internet offers global connectivity, vast information resources, real-time communication, e-commerce, education, business opportunities, entertainment, and innovation.
- **Disadvantages of the Internet:** Concerns related to privacy, cybersecurity risks, information overload, digital addiction, the digital divide, online harassment, and health impacts are associated with the Internet.
- **Common Applications of the Internet:** The Internet is used for communication, information retrieval, e-commerce, social media, entertainment, education, business, research, and innovation.

National Book Foundation

Exercise

Select the suitable answer for the following Multiple choice questions.

i. What computing machine was capable of taking input from punch cards and storing data in memory?

a) Abacus
b) Analytical Engine
c) Tabulating Machine
d) Differential Analyzer

ii. What technology replaced vacuum tubes in second-generation computers?

a) Transistors
b) Integrated Circuits (Ics)
c) Microprocessors
d) Magnetic tapes

iii. Which computer generation introduced the use of Integrated Circuits (Ics)?

a) First Generation
b) Second Generation
c) Third Generation
d) Fourth Generation

iv. Which of the following is an example of a natural system?

a) Telephone network
b) Human body
c) Database
d) Automobile

v. Which component of the microprocessor controls the working of input/output devices and storage devices?

a) Arithmetic Logic Unit (ALU)
b) Control Unit (CU)
c) Memory Address Register (MAR)
d) Data Register (DR)

vi. Which type of memory retains data even when the computer is turned off?

a) RAM
b) ROM
c) Cache
d) Register

vii. What is the main purpose of Cache memory?

a) Store program instructions
b) Provide high-speed storage
c) Control input/output devices
d) Perform arithmetic operations

viii. What is the smallest unit of memory in a digital computer?

a) Kilobyte
b) Bit
c) Byte
d) Megabyte

ix. Which memory type uses laser beams to read and write data?

a) RAM
b) Optical memory
c) Cache memory
d) ROM

National Book Foundation

x. In Von Neumann architecture, where are instruction data and program data stored?

a) In separate memory units
b) In the same memory unit
c) In the processor's registers
d) In the cache memory

xi. What is the primary function of the buses in a computer's memory system?

a) Control memory operations
b) Display graphics
c) Transmit data between components
d) Perform arithmetic operations

xii. What is the primary function of the Presentation Layer in the OSI model?

a) Establishing connections between computers
b) Converting data into a standard format
c) Deciding the physical path of data
d) Sending data in both directions simultaneously

xiii. In which network topology is data transmission unidirectional, like radio or television broadcasts?

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

xiv. Which data communication standard is used for transferring web pages and related data on the World Wide Web?

a) TCP/IP
b) HTTP
c) FTP
d) SMTP

xv. Which network architecture requires each computer to act as either a server or a client but not both simultaneously?

a) Client/Server Network
b) Peer-to-Peer Network
c) Metropolitan Area Network (MAN)
d) Wide Area Network (WAN)

Give Short answers to the following short response questions (SRQs).

i. What is the function and significance of Napier's Bone in early computing.
ii. How did Charles Babbage's Difference Engine differ from the Analytical Engine?
iii. What is the primary goal of fifth-generation computers?
iv. Name the three types of memory mentioned on the motherboard in the Von Neumann architecture?
v. Differentiate between volatile and non-volatile memory.
vi. Show memory hierarchy with the help of diagram.
vii. Why is the word size of a processor important, and how does it affect the computer's performance?

National Book Foundation

viii. What is pipelining, and how does it improve the efficiency of a CPU?

ix. Imagine you are building a custom computer for a friend. Give three considerations you would take into account when choosing the type and size of memory for this computer.

x. You are a computer technician tasked with upgrading an older computer with a slow CPU. How might you improve its performance without replacing the entire CPU?

xi. Your school is planning to purchase new printers for the computer lab. Give three advantages and disadvantages of impact and non-impact printers to help them make a decision.

xii. Define "protocol" in the context of data communication.

xiii. Define simplex, half-duplex, and full-duplex modes of communication with one example each.

xiv. What are start and stop bits, and where are they used in data transmission?

xv. How does a switch differ from a hub in a network?

xvi. How does asynchronous transmission differ from synchronous transmission, and in what situations are they typically used?

xvii. Explain the concept of a peer-to-peer network and its limitations compared to a client/server network.

xviii. Compare and contrast the star and ring network topologies.

xix. Why is mesh topology considered the most reliable but also the most expensive to implement?

xx. What is the role of the Application Layer in the OSI model?

Give Long answers to the following extended response questions (ERQs).

Q1. Describe the evolution of computer generations from the first generation to the fourth generation, highlighting the key technological developments and their impact on computing.

Q2. Discuss the importance of categorizing and understanding different types of systems, both natural and artificial, and provide examples of each type.

Q3. Explain the characteristics and potential challenges associated with fifth-generation computers that aim to understand natural languages and possess thinking capabilities. What are the implications of such advancements in computing on society?

Q4. Explain the fundamental components of network communication, and how do they work together to facilitate data transfer?

Q5. Describe the roles of common communication devices like hubs, switches, routers, and gateways in data communication. How do they contribute to the functionality of a network?

National Book Foundation

Q6. Discuss the advantages and limitations of different network topologies, including bus, star, ring, and mesh. When should each topology be used in a network design?

Q7. What is the OSI model, and how does it help in understanding the process of data communication? Explain each of the seven layers and their functions.

Q8. Explain the evolution of the Internet from its origins to the modern-day global network. What major technological advancements contributed to its growth?

Q9. Discuss the advantages and disadvantages of the Internet, considering factors like global connectivity, information access, privacy concerns, and digital addiction.

Q10. Explore common applications of the Internet and their impact on various aspects of society, including communication, education, business, entertainment, and research.

Lab Activities

Activity 1

Computer History Timeline Presentation:

Objective: To help students understand the historical development of computer systems.

Instructions:

i. Divide the class into small groups (5 to 6 students) and assign each group a specific era or generation of computers (e.g., First Generation, Second Generation, etc.).

ii.` Instruct each group to create a presentation (using PowerPoint or other presentation tools) that highlights the key inventions, innovations, and characteristics of their assigned era.

iii. Encourage students to include images, diagrams, and historical context to make their presentations engaging and informative.

iv. Have each group present their findings to the class, creating a comprehensive timeline of computer history from early computing devices to modern computers.

Outcome: This activity will help students visualize the progression of computer technology over time and understand the significance of each generation's contributions.

National Book Foundation

Activity 2

Natural vs. Artificial Systems Comparison:

Objective: To differentiate between natural and artificial systems based on the text's definitions and examples.

Instructions:

i. Provide students with the definitions and examples of natural and artificial systems from the text.

ii. Ask students to work individually or in pairs to research and identify additional examples of each type of system beyond those mentioned in the text.

iii. Have students create a visual comparison chart or poster that categorizes these examples into natural and artificial systems. They should explain why each example belongs to its respective category.

iv. Encourage a class discussion where students share their findings and insights.

Outcome: This activity promotes critical thinking and a deeper understanding of the concepts of natural and artificial systems.

Activity 3

Computer System Components Showcase:

Objective: To familiarize students with the core components of a computer system.

Instructions:

i.` Set up a "Computer System Components Showcase" in your classroom with physical representations of computer components, such as a motherboard, CPU, keyboard, mouse, microphone, scanner, and other input devices.

ii. Provide students with a checklist or worksheet that lists these components.

iii. Divide the class into small groups and assign each group to visit the showcase. Their task is to identify and describe the purpose of each component.

iv. Afterward, reconvene as a class and have each group share their observations and descriptions.

v. Discuss the role of each component in a computer system and how they work together.

Outcome: This hands-on activity enhances students' understanding of computer hardware components and their functions in a computer system.

National Book Foundation

Activity 4

Create a Data Communication Network

Objective: To understand the components and types of data communication networks.

Instructions:

i. Divide the students into groups and provide each group with a set of components mentioned in the text (Sender, Message, Medium, Protocol, Receiver).

ii. Instruct each group to create a visual representation of a data communication network using these components. They can use drawings, diagrams, or digital tools.

iii. Ask each group to label the components and explain their roles in the network.

iv. Have each group present their network to the class, highlighting the type of medium they've chosen (wired or wireless).

v. Discuss the advantages and disadvantages of using different types of mediums in data communication.

Activity 5

OSI Model Role-Play

Objective: To understand the layers of the OSI model and their functions.

Instructions:

i. Explain the OSI model to the students, emphasizing its seven layers.

ii. Assign each student or pair of students a specific layer from the OSI model.

iii. Ask the students to prepare a short role-play or skit in which they act out the responsibilities and interactions of their assigned OSI layer.

iv. Encourage creativity in demonstrating how each layer interacts with the others.

v. Have the students perform their role-plays in front of the class.

vi. After each role-play, discuss how the layers work together to ensure successful data communication.

National Book Foundation

UNIT 2

Computational Thinking & Algorithms

After completing this lesson, you will be able to:

- understand and apply techniques to decompose problems.
- solve simple and complex problems computationally.

National Book Foundation

UNIT INTRODUCTION

A computer is a machine that is used to solve problems by accepting inputs, performing operations, and presenting outputs. Computers can't think but can perform those operations that are written by us in the form of algorithms and fed into the computer in the form of programs. You can't program a computer to do something that you don't know how to solve. Therefore, you first need to understand the real-world problem, make some solution, and instruct the computer to behave accordingly when it is asked to solve that problem. Although computers are of various types depending upon their size and capacity but computation is primarily based on processor and main memory. We use computers because they work at a much faster speed and perform operations more accurately than humans.

2.1 Problems

A problem is a challenge or situation that needs to be overcome using some action. The problems are present in all fields such as economic, healthcare, education, transportation, internet, biology and many more.

Ÿ What to produce? Ÿ How to manage limited budgets?	Ÿ How to expand access to medicines? Ÿ How to stop infectious diseases?	Ÿ How to increase literacy rate in Pakistan? Ÿ What subject areas of education should be focused on more?
Economic	**Healthcare**	**Education**

National Book Foundation

Ÿ Why is the number of vehicles growing up? Ÿ Why the accidents on motorways are increasing?	Ÿ How to secure mobile banking transactions? Ÿ Why do websites show unwanted advertisements?	Ÿ How do 5G cell phone signals harm humans? Ÿ How do trees give earth all its oxygen?
Transportation	**Internet**	**Biology**

2.2 Identifying a Computing Problem

In computer science a problem is a challenge or situation that needs to be overcome using some action. In computer science, a computing problem is a problem that is solved step-by-step using computation. It can include any type of calculation such as arithmetical or logical. These problems usually have a well-defined input and some desired properties that output must be satisfied. Following are some of the computing problems:

- Decision Problems
- Search Problems
- Counting Problems

Decision Problems: A decision problem occurs when a given input requires a binary response, either Yes or No. Responses may take various forms, such as true or false. Figure 2.1 visually illustrates a simple decision problem. In complex cases, answers extend beyond a straightforward Yes-or-No, involving multiple decision factors and criteria.

The examples of simple decision problems are:

- The problem whether a given number is odd (or even).
- The problem whether a given number is a prime number.
- The problem that is there any occurrence of "aa"?" in a sequence *x* of English alphabets.

National Book Foundation

Search Problems: In science and engineering, many problems are solved using the search. In such types of problems, we have a set of objects among which we search for the solution. For example, finding a path between two cities. Search problems are often represented using graphs. Where we have nodes and where each link connects to nodes. A node can be connected to multiple nodes. To solve such problems, we have three things:

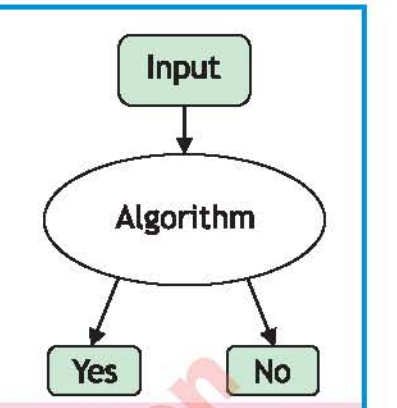


Fig. 2.1: Decision Problem

- *Initial State:* represents the node from where we need to start search.
- *Operations:* represents the moves that transition from one node to another.
- *Goal:* defines the target of end condition.

Example: Route Finding Problem

The map shown in Fig. 2.2 corresponds to the graph, where the nodes are assumed as cities and the links between nodes represent the routes in between the cities. The problem is to find a route in the graph from city S to city T.

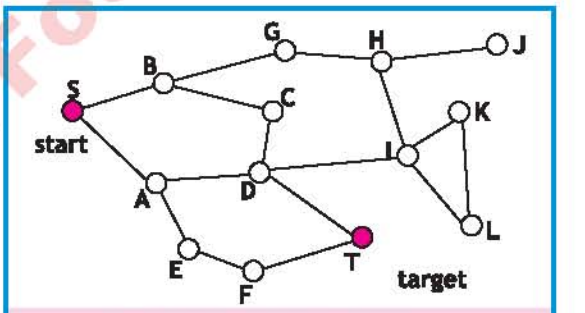


Fig. 2.2: Route Finding Problem

Example: Eight puzzle problem

A 3×3 board is given that contains 8 tiles (every tile can contain one number ranging from 1 to 8) and one empty space. The goal is to place the numbers on tiles in such ordering that it matches the final configuration using the empty space. However, we also have some restrictions that we can only slide one step and only four types of slides are allowed (left, right, above, and below).

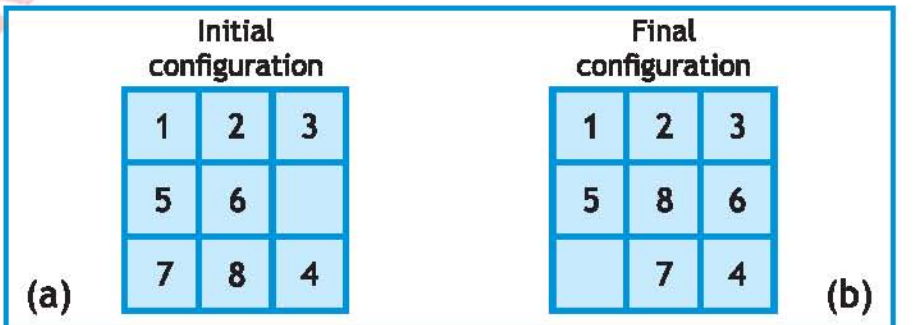


Fig. 2.3: Eight puzzle problem

National Book Foundation

DO YOU KNOW

Have you used Google Map application?
It is a search problem.

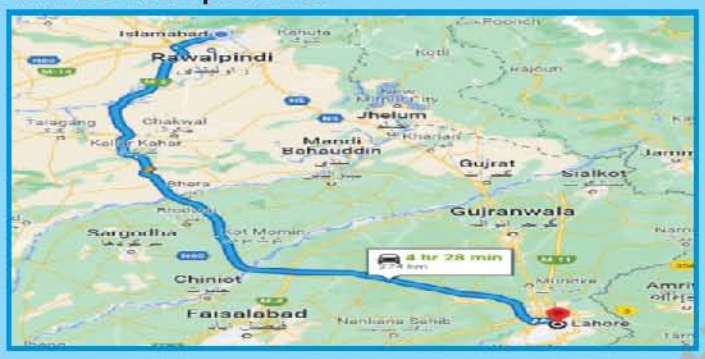


Fig. 2.4: Google Map

Counting Problems: These problems work on the principle that if an event/decision has A number of choices and another decision/event has B number of choices then the total number of possible unique combinations would be A x B.

Let's solve counting problems.

Example: One event is numbers of shirts, and another event is number of pants you own then how many pairs of shirts and pants you can make from it.

Fig. 2.5: Counting Problem Example 1

Solution

Total possible casual dresses = (number of shirts) x (numbers of pants).

Total casual dresses = 5 x 3 = 15

National Book Foundation

The counting problems are normally seen as an easy task that can immediately be done. As we go deeper, the counting problems can get complicated very quickly if we want to list out all of the possibilities.

Fig. 2.6: Counting Problem Example 2

Example:

You visit a computer shop to buy a computer system for you. The vender asks you to choose one of 4 monitors, one of 2 keyboards, one of 4 computers and one of 3 printers. How many numbers of possible systems you can choose from?

Solution

You must choose 1 monitor, 1 keyboard, 1 computer and 1 printer. The below given diagram shows each component of the computer system with the number of choices you have.

For the computer system elements, we have 4 monitors, 2 keyboard, 4 computers and 3 printers. Using the counting principle, the number of all possible computer systems that you can buy is given by:

$$N = 4 \times 2 \times 4 \times 3 = 96$$

2.3 Problem Solving

Problem solving defines the process of analyzing some situation and accordingly behaving to generate some response. For simple problems the following 4 steps are used:

a. Define and analyze a problem: What is the problem and why it is happening.
b. Design a plan: What we are going to do (Algorithm)
c. Implement the plan: Code it by using some programming language.
d. Evaluate: did our plan work?

However, for complex problems, the 6 step problem solving process could be used

a. Define and analyze a problem
b. Decompose the problem: make sub-problems that are manageable
c. Identify potential plans for each sub-problem: such as plan A, plan B and plan C
d. Select and design best plan
e. Implement that plan
f. Evaluate

Before jumping into an action for a problem, it must be thoroughly investigated to capture all the relevant aspects of the problem. For this purpose, we properly define and analyze a problem. In defining a problem, there must be clarity and the objectives of the problem must be specified.

National Book Foundation

2.4 Input-Processing-Output (I-P-O) Model

Fig. 2.7: IPO Model

The computer systems work on the input-Process-Output model. The **analyzing of problem allows us** to break the problem down into three components: Input, Output and Processing. it requires us to have correct identification of:

- input that needs to be given to the system,
- operations the system would perform and
- the output to be presented.

Input refers to the requirements from the environment, operations are the computation that is dependent upon the requirement while output is the thing that is presented to the environment.

2.5 Input-Processing-Output (I-P-O) Chart

To visually represent the IPO model, we use IPO charts that represent the Inputs, Process and Output in tabular form. IPO charts are considered a handy tool that software designers use to solve problems.

Input	Process	Output
Data entered in the system	Operations that will be applied on the input	Data that has been turned into the Information

For example, an IPO chart to compute the area of rectangle would be:

IPO chart to compute area of Rectangle

Input	Process	Output
Length Width	Area = Length * Width	Area

Fig. 2.8: IPO Chart

National Book Foundation

The input column of the above IPO chart is the input to the system. In this particular example, we need the *length* and the *width* of the rectangle to compute the area.

Process column contains the processing that the system will perform on the input to provide generate information. So, to compute the rectangle area we need to perform the multiplication operation on the input e.g. the length and the width of the rectangle.

*area= length * width*

The output provides the information to the user. Therefore, the information for the user is the area of the rectangle.

Example

Calculate a student's average based on his grades.

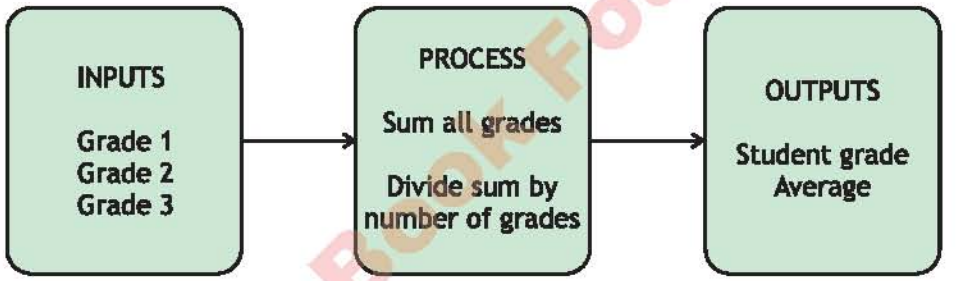


Fig. 2.9: IPO Chart Example

2.6 Computational Thinking

Using computation to solve problems requires the ability to think in a certain way, which is often referred to as 'computational thinking'.

Tip

Before You Learn Programming, Understand Computational Thinking.

2.6.1 Importance of Computational Thinking

Computational thinking encourages not only to reflect clearly on a problem they're solving but also to define a repeatable solution for it. While solving a problem, we don't just want to solve one instance of a problem, we want an automized system that can solve all instances of similar problem. Computational thinking can be thought of as an extension of logical thinking.

National Book Foundation

2.6.2 Properties of Computational Thinking

- **Decomposition:** Breaking down the larger problems into smaller/ manageable ones and working on them one by one. These smaller problems are referred as sub-problems. This way we simplify the problem and solve it easily.
- **Abstraction:** by removing the unnecessary details to solution, so you could be able to identify essential information.
- **Pattern Recognition:** Examine the problem for a pattern or similarities between previously solved problems.
- **Algorithm Design:** This is actual designing of solution. This involves creating step-by-step plan of the problem solution.

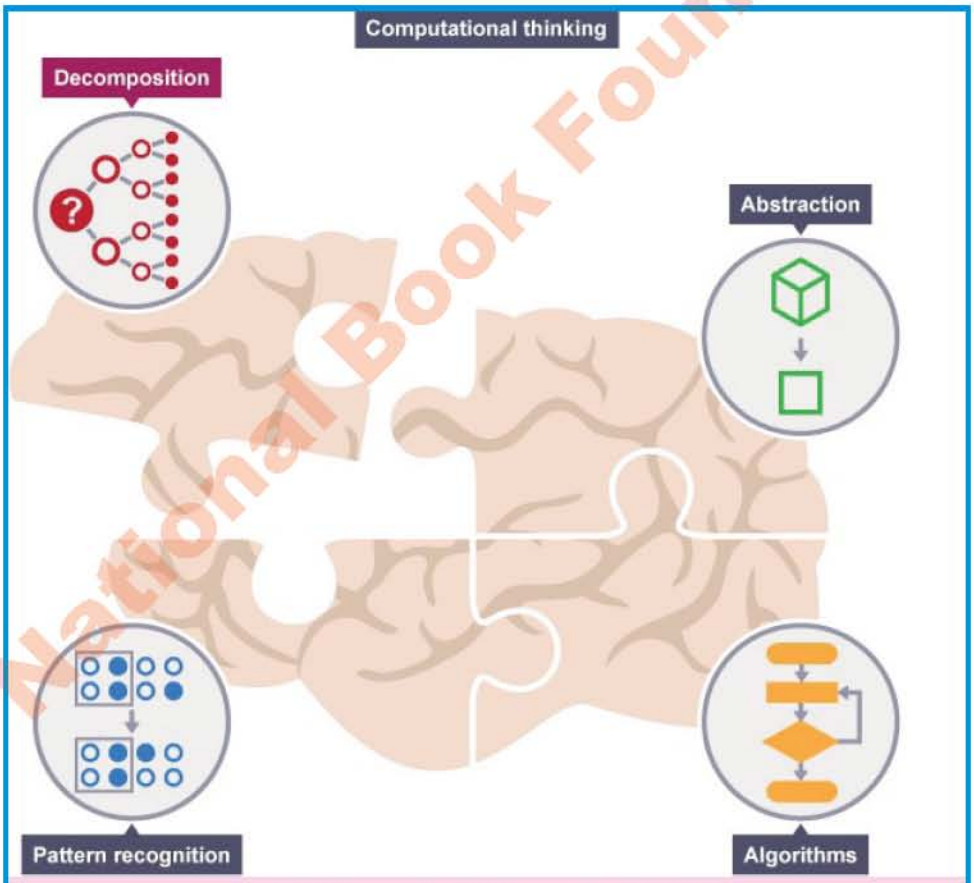


Fig. 2.10: Properties of Computational Thinking

National Book Foundation

Example: If you want to create your own computer game, how these properties of computational thinking would apply:

Consider the following diagram.

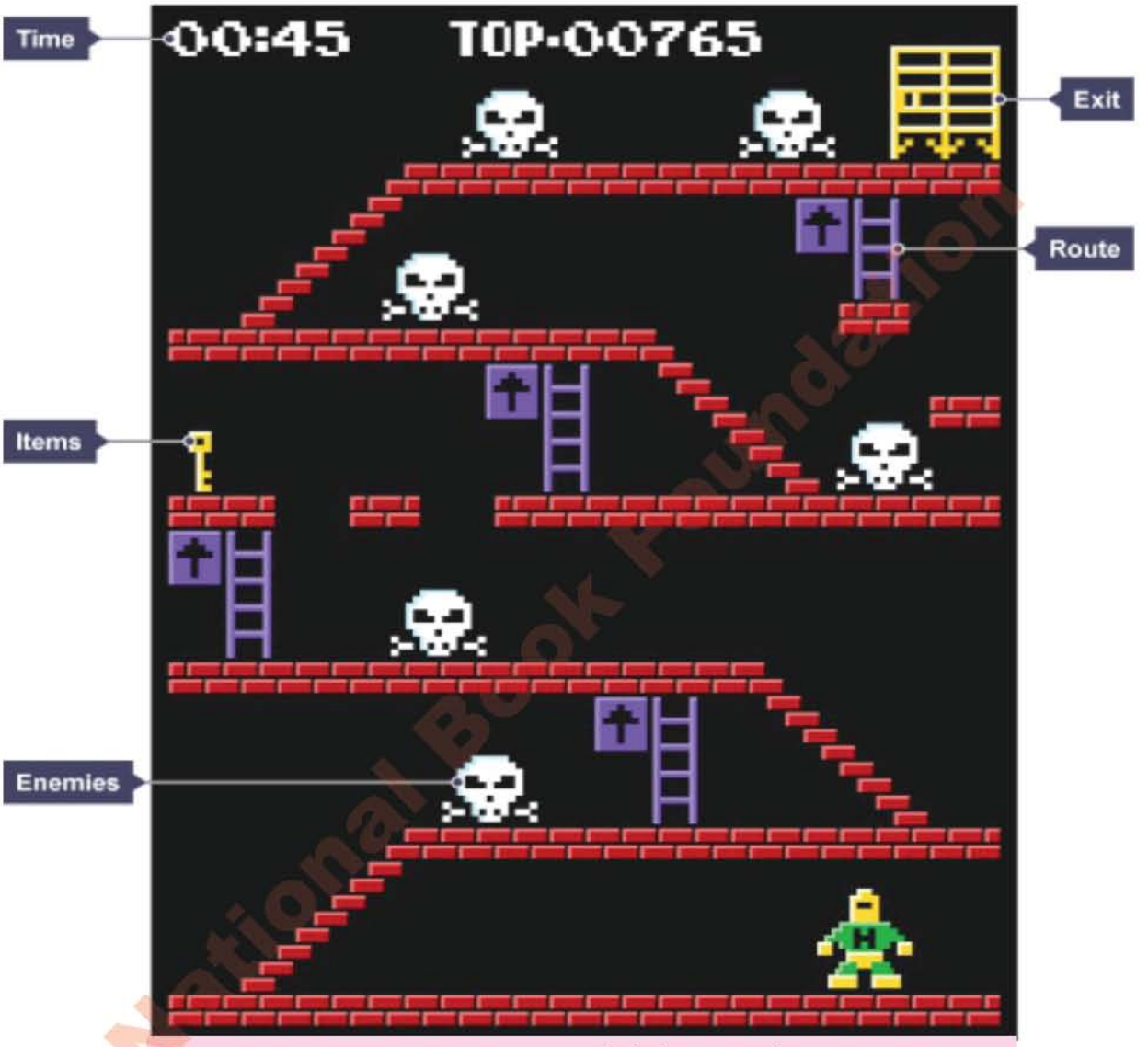


Fig. 2.11: Computational Thinking Example

Decomposition: where to go, how to complete the level

Abstraction:

Necessary Information: location of exit, where is enemy, etc

Unnecessary Information: weather

Pattern recognition: Six enemies should be handled like a single enemy.

Algorithm Design: step by step plan of action e.g. movement

National Book Foundation

2.7 Principles of Computational Thinking

2.7.1 Logical thinking

Logical thinking refers to analyzing a particular situation or problem using reason and accordingly reaching up to one or more decisions that are sensible. The situation analysis requires gathering of relevant facts and then deciding the best way based on the reasons.

For example, you enter your home and see water on the floor. What do you think has happened?

To solve such problems, you need reasoning skills that are based on evidence and facts.

Example:

Fact: Islamabad is a city
Fact: Cities have a mayor (or some equivalent)
Inferred: Islamabad has a mayor (or some equivalent)

One of the fundamental parts of computational thinking is to think logically. The computers use logic in their computation but it does not mean that they think logically. The computers themselves can't perform logical thinking unless they are programmed to do so. By developing the ability to align problem-solving with technology and focusing on the process as well as the solution

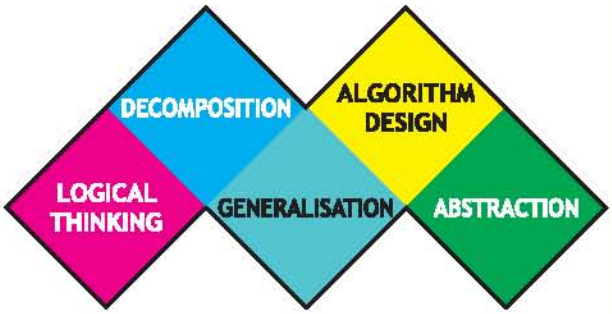


Fig. 2.12: Logical thinking and Computational thinking

National Book Foundation

2.7.2 Algorithmic Thinking

Algorithmic thinking is about problem-solving. Computers do nothing by magic, algorithmic thinking is a way of getting to a solution. It is a process that involves identifying the steps needed and then implementing those steps in a logical and efficient manner. It is a way of thinking that involves breaking down complex problems into smaller, manageable parts and then solving those parts one at a time.

By developing algorithmic thinking skills, you can become a better problem solver.

Example: Find the largest of three unequal numbers.

Let's first draw IPO chart of the problem.

Input	Process	Output
Three numbers	Find largest among the three numbers	Display largest number

Now we write algorithm of the process part of IPO.

Step 1: Let three number be A=10, B=20 and C=30

Step 2: Check if A is the largest?

Step 2.1: Check if A > B and A > C then A is the largest.

Step 3: If A is largest then Stop the process, otherwise proceed to next step.

Step 4: Check if B is largest?

Step 4.1: Check if B > A and B > C then B is the largest.

Step 5: If B is largest then Stop the process, otherwise proceed to next step.

Step 6: C is largest.

2.8 Methods to Design a Solution

Once you have completely understood the problem, the next stage is to design a solution. In solution design, you define how a software will meet the requirements and objectives of a problem.

There are two methods that are used to design a solution:

- Flowcharts
- Concept Maps

Let's discuss both in detail and solve some examples.

National Book Foundation

2.8.1 Flowcharts

Flowchart is a diagrammatic representation of an algorithm. It describes what operations are required to solve a given problem.

Importance of flowchart in solving a problem

Flowchart illustrates the sequence of operations to be performed to solve a problem in the form of a diagram. Computers programmers draw flowcharts before writing computer programs. It provides an easy way to analyze and find solutions to problems. Once the flowchart is drawn, it becomes very easy to write the program in any computer language. It is very helpful in communicating the problem-solving method to other people. It also helps in finding and removing logical errors.

Steps for drawing flowchart

The flowchart developer must determine the following requirements for the given problem or algorithm before drawing a flowchart.

- Start of the flowchart
- Input to the flowchart
- Type of processing required
- Decision to be taken
- Output of the operation
- End of the flowchart

Start of the flowchart: Every flowchart should start with a trigger. The graphical representation of start of flowchart is oval shape.

Input to the flowchart: The flowchart designer must know what exactly the input to the flowchart is. The input is determined from the problem statement. For example, the given problem is to convert temperature from Fahrenheit to Celsius. Here, the input will be the temperature in Fahrenheit. The graphical representation of input is parallelogram.

Processing in the flowchart: The flowchart designer must decide what type of calculation is to be performed or what formula is applied to obtain the required result. For example, to find the area of a triangle. The following formula is to be used:

Area = (Base x Height) / 2

The graphical representation of processing symbol is rectangle.

National Book Foundation

Decision making in flowchart: The flowchart designer must decide which control structure (sequence, repetition, or selection) are to be applied for the solution of the problem. For example, selection structure must be applied to print letter grade of a student based on the marks obtained. The selection structure will check in which range the marks fall and accordingly print the grade. Graphically the decision is represented with diamond.

Output: The flowchart must provide the required output. The output is represented with a parallelogram symbol as well.

End of flowchart: The flowchart should have a defined end, and because of the possibility of multiple decision points, it may have multiple ends. For end of flowchart the oval symbol is used.

National Book Foundation

Flowchart to solve problems:

Flowchart 1: Flowchart to find sum, product, and average of five numbers.

From the statement of the flowchart, it is clear that sum, product and average are to be calculated and given are five numbers. The flowchart is shown below for this.

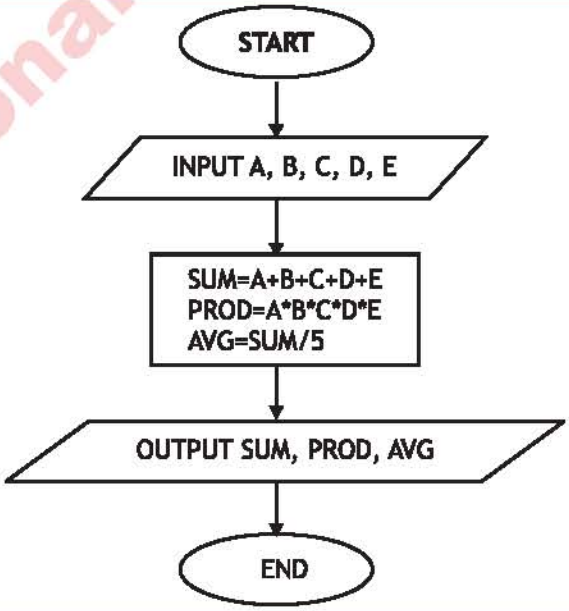


Fig. 2.13: Flowchart to find sum, product, and average

Flowchart 2: Flowchart to find the largest of three unequal numbers.

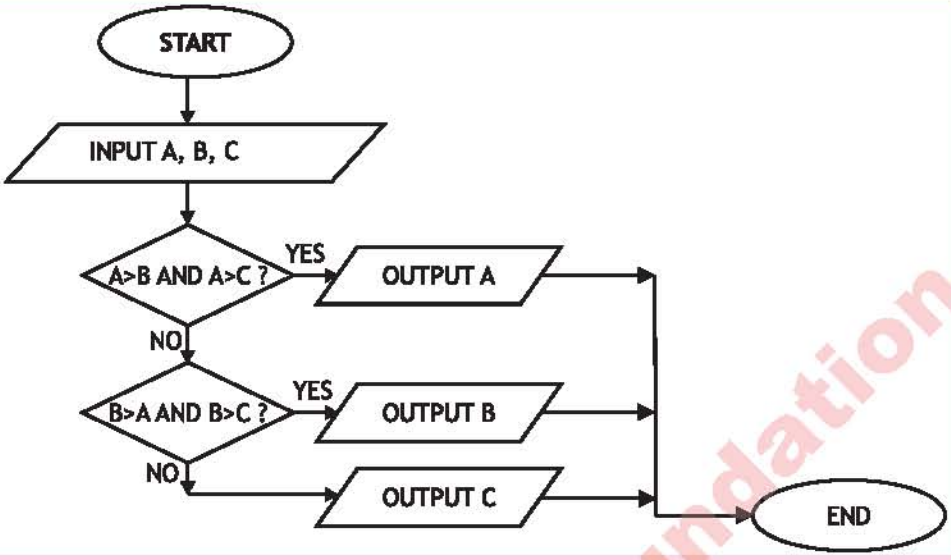


Fig. 2.14: Flowchart to find the largest of three unequal numbers

Flowchart 3: Flowchart to find the area of triangle when the lengths of height and base are given.

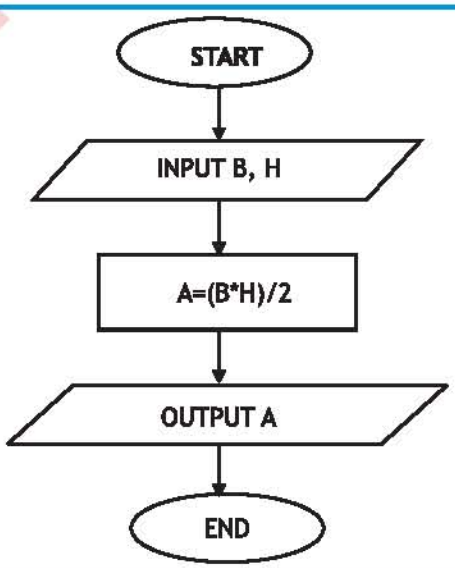


Fig. 2.15: Flowchart to find the area of triangle

National Book Foundation

Software Tools for Flowchart Designing

For flowchart designing, different software tools are available to design the flowcharts. Some of the famous tools are Microsoft Visio and LARP software.

Microsoft Visio

Microsoft Visio is a tool for drawing various types of diagrams such as flowcharts, building plans, data flow diagrams, network diagrams, etc.

How to create flowchart in Microsoft Visio

a) Start Microsoft Visio
b) Click on the category of Flowchart
c) Double-click the Basic Flowchart
d) For each step, you wish to design in the process, drag a relevant flowchart symbol and place it onto your drawing.
e) Connect the flowchart shapes by holding the mouse pointer over the first symbol, and then releasing it on the to the other symbol you wish to connect to.
f) To place text into a shape, select it, and then type.

The Figure 2.16 shows a screenshot of flowchart designing in Microsoft Visio.

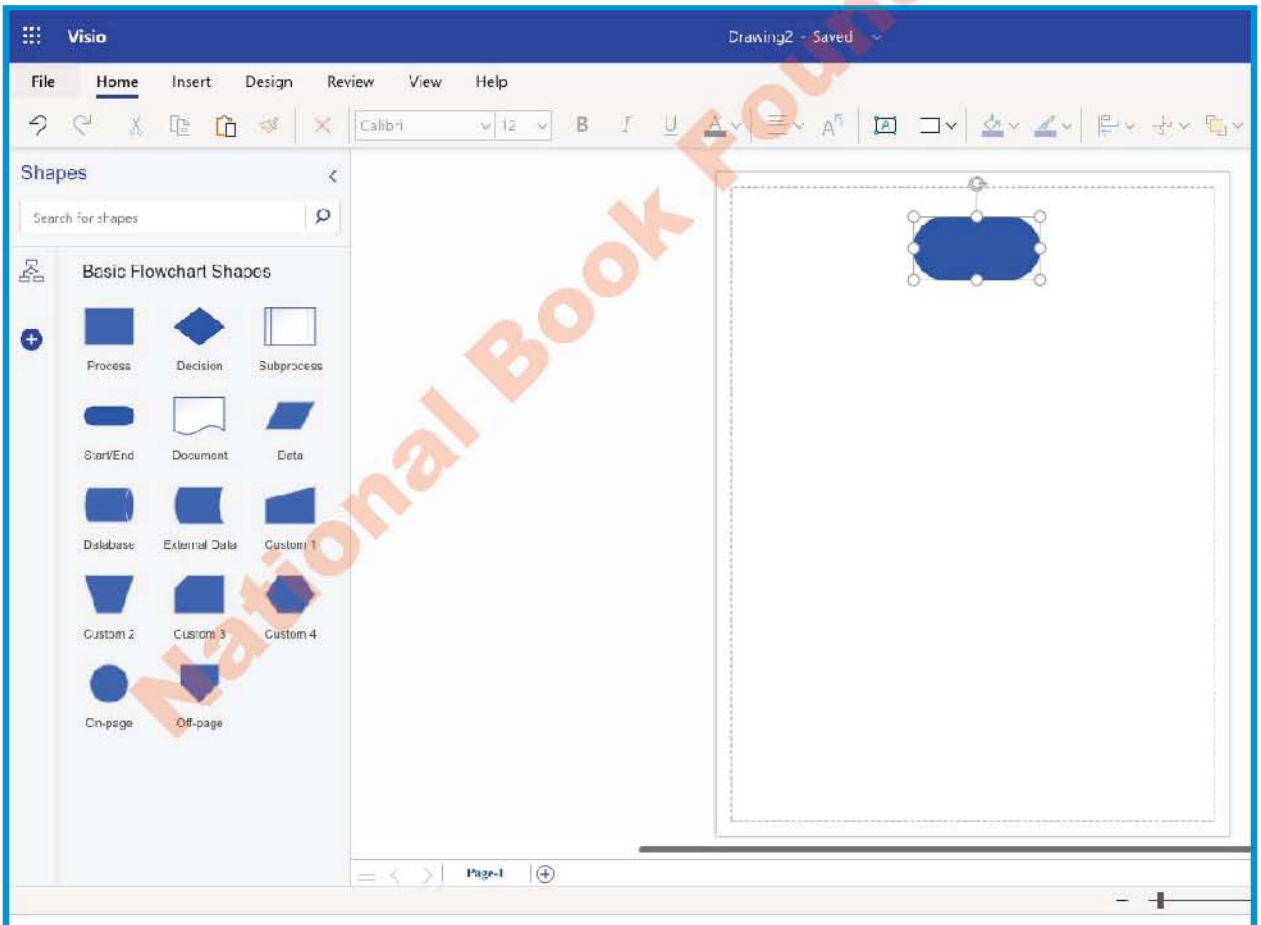


Fig. 2.16: Screenshot of Microsoft Visio for designing flowchart

National Book Foundation

LARP

Logics of Algorithms and Resolution of Problems (LARP) is a programming language for rapid development of prototypes. However, unlike C++, Java and other programming languages, LARP uses semi-natural syntax. Therefore, it is very easy for non-programmers to understand. LARP also allows algorithms to be expressed as flowcharts. A screenshot of flowchart designing is shown in Figure 2.17.

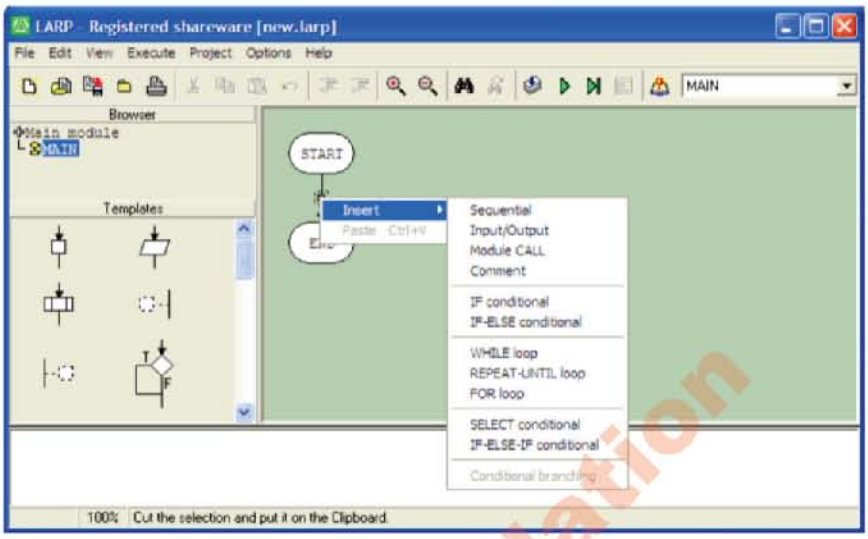


Fig. 2.17: Screenshot of LARP software

2.8.2 Concept Maps

Like a flowchart, a concept map is another way of representing knowledge.

A concept map is a graphical tool that represents concepts and relationships between

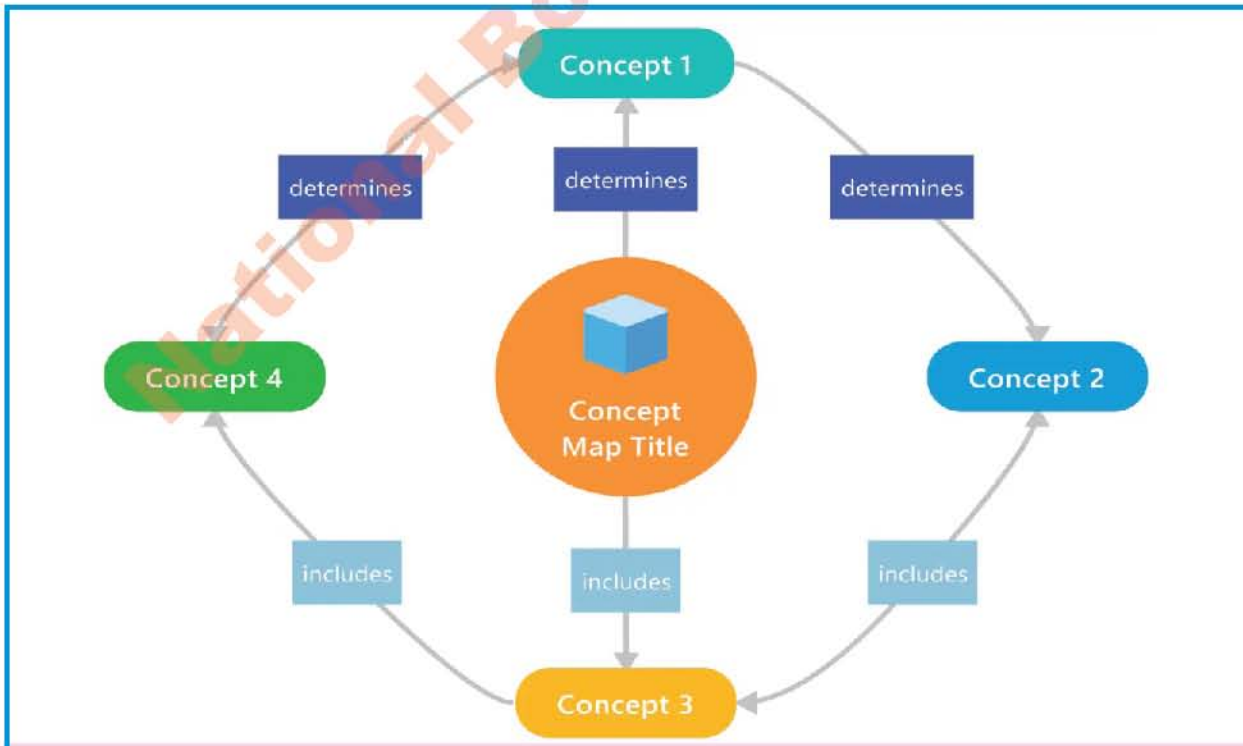


Fig. 2.18: Concept Maps

National Book Foundation

them. The concepts in these maps are represented as boxes or circles, which relate to lines or arrows. These lines are labeled with linking words and phrases to represent the connections between concepts. There are different software tools for concept mapping, for example CmapTools, Mind Manager. (Fig 2.18 shows an example concept of map)

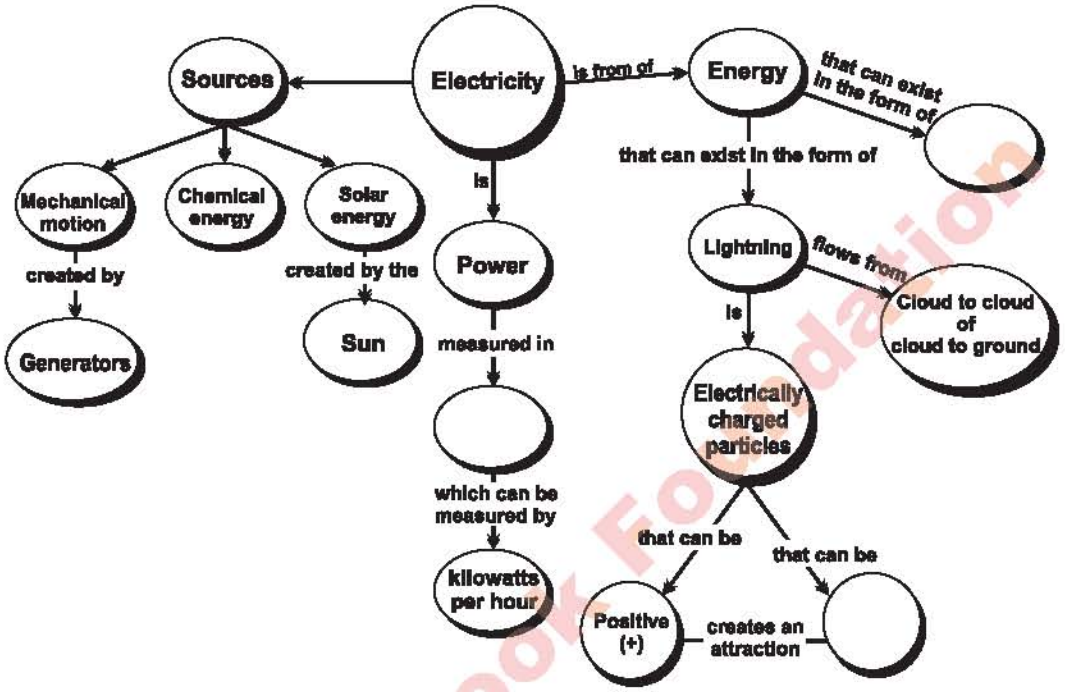


Fig. 2.19: Concept Maps Example

Example: Consider the below given sample diagram representing concept map of electricity

National Book Foundation

Summary

- **Abstraction:** Identifying essential information and Removal of the unnecessary details to solution.
- **Algorithm Design:** This is actual designing of solution. This involves creating step-by-step plan of the problem solution.
- **Algorithmic Thinking** is a way of getting to a solution.
- **Computing Problem** is a problem that is solved step-by-step using computation.
- **Computational Thinking** is a problem-solving method that computer scientists use as a skill to solve complex problems in a logical and systematic manner.
- **Concept Map** is another way of representing knowledge.
- **Counting Problems:** These problems work on the principle that if an event/decision has A number of choices and another decision/event has B number of choices then the total number of possible unique combinations would be A x B.
- **Decision Problems:** A decision problem is the situation for a given input that has Yes-or-No Answer.
- **Decomposition:** Breaking down the larger problems into smaller/ manageable ones and working on them one by one. These smaller problems are referred as sub-problems. This way we simplify the problem and solve it easily.
- **Flowchart** is a diagrammatic representation of an algorithm.
- **IPO charts** are considered a handy tool that software designers use to solve problems.
- **LARP:** Logics of Algorithms and Resolution of Problems is a programming language for rapid development of prototypes
- **Logical Thinking** refers to analyzing a particular situation or problem using reason and accordingly reaching up to one or more decisions that are sensible.
- **Pattern Recognition:** Examine the problem for a pattern or similarities between previously solved problems.
- **Problem** is a challenge or situation that needs to be overcome using some action.
- **Problem Solving** defines the process of analyzing some situation and accordingly behaving to generate some response.

National Book Foundation

Exercise

Select the suitable answer for the following multiple-choice questions (MCQs).

1. For a problem, we face in real world situations. In what sequence we follow the steps.
   I. make some solution.
   II. understand the real-world problem.
   III. instruct the computer to behave accordingly.
   a) I, II, III b) I, III, II c) II, I, III d) II, III, I
2. Following are types of computing problems
   I. Counting Problems
   II. Search Problems
   III. Decision Problem
   a) I and II b) I and III c) II and III d) I and II and III
3. Computational thinking is
   a) Programming b) Thinking like a computer
   c) Coding d) Logically solving problems
4. To solve Search problems, we need to
   a) Provide the moves. b) Provide start state.
   c) Provide the end state. d) a, b and c
5. The eight queens puzzle is the problem of
   a. Sorting b. Searching
   c. Counting d. Both a and b
6. Finding the location of the element with a given value is
   a) Search b) Traversal
   c) Sort d) None of above
7. In IPO Charts, we have
   a) Input, Plan, Output b) Input, Program, Output
   c) Input, Process, Output d) Input, Proceed, Output

National Book Foundation

Give Short answers to the following short response questions (SRQs).

1. What is the major difference in solving simple problems and complex problems?
2. Why software designers prefer to use IPO charts?
3. Differentiate between Computational thinking and Logical thinking.
4. Write four properties of Computational thinking.
5. What are the methods used to design a solution?
6. Which Computational thinking technique breaks down the problem into smaller parts?
7. Identify 3 computing problems from other subjects that you are studying in your class.
8. Why do we need to think computationally?
9. The telephone numbers usually have 9 digits. Out of these 9, the first two digits represents the area code and are it remained constant within a given area. The last 7 digits represents the number, and it cannot begin with 0. How many different telephone numbers are possible with a given area code.
10. From city A to city B, there are 4 different roads and from city B to city C there are 2 different roads. Draw a map of given situation and identify how many possible routes are there that someone can follow to reach from city A to city C passing by city B?

Give Long answers to the following extended response questions (ERQs).

Q 1. Identify whether the given problems are Decision Problem, Counting Problem or Search Problem. Write your answer in front of each problem given below:

a. Does a given binary string have an even number of zeros?
b. Flipping a coin result in Head or tails. I flip a coin 20 times, how many different sequences of heads and tails are possible?
c. Does a certain Java program say "yes" to an empty input?
d. How many ways can the letters of the word TRIANGLE be arranged?
e. N-queens problem: where the goal is to place eight queens on a chessboard such that no queen attacks any other.

Q 2. A student has to take one course of physics, one of science and one of mathematics. He may choose one of 3 physics courses (P1, P2, P3), one of 2 science courses (S1, S2) and one of 2 mathematics courses (M1, M2). In how many ways can this student select the 3 courses he has to take?

Q 3. Create an IPO chart which will accept the ages of four boys and calculate their total age and average age. The program must display both the total age and the average age.

National Book Foundation

Q 4. Create an IPO chart of a scenario that allows a user to enter in two numbers. The operation to be performed is either addition, subtraction, multiplication or division and accordingly the output should be given to the user.

Q 5. The child wants to plan a birthday party for their friend.

a) Draw an IPO chart of this situation.

b) Write down properties on computation thinking.

LAB ACTIVITIES

Activity 1

Draw a flowchart in MS Visio that takes input of two number A and B and output TRUE if A is greater than B otherwise it should output FALSE.

Activity 2

Draw IPO chart and algorithm for the following:

a) Find the exponent of a given number: Exponent or power of a number means how many times to use the number in a multiplication. In other words, it is the product of a number that is multiplied as many times as the exponent.

b) Print odd numbers from 1 to 100. Such as 1 3 5 7 9 11 . . . 99

c) Print the following sequence of numbers in descending order.

27 24 21 18 15 12 9 6 3 0 3 6

d) Find the sum of even numbers up to 100.

SUM = 2 + 4 + 6 + 8 + 10 + 12 + 14 + . . . + 100

e) Print a multiplication table of a given number.

Activity 3

Convert the algorithms of Lab Activity given in Q 2 to flowcharts.

Teacher's Guide

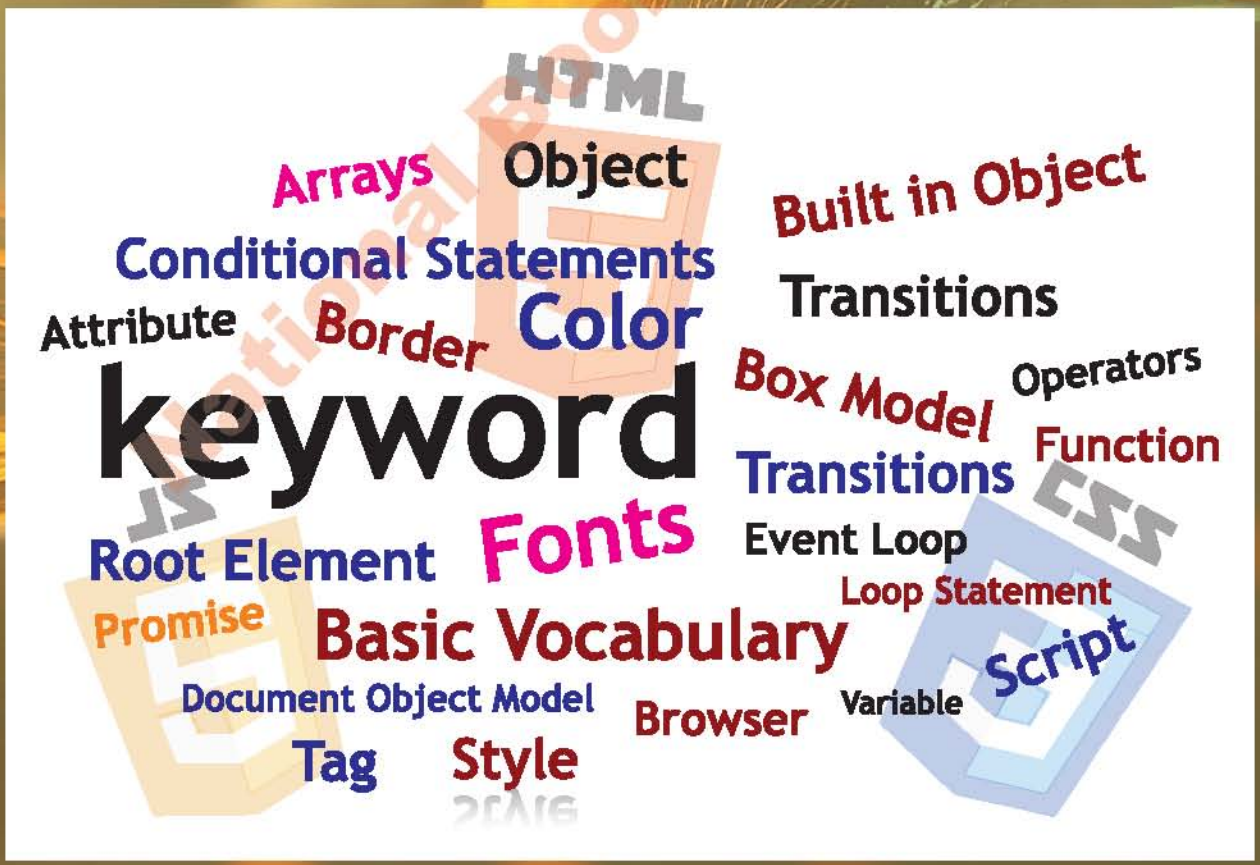
A thorough framework for using computational thinking techniques in the classroom is offered by the International Society for Technology in Education (ISTE) Standards. It highlights that educators ought to be capable of "develop algorithms that break down problems into smaller, more manageable subproblems." (https://iste.org/standards/computational-thinking-competencies)

National Book Foundation

Programing Fundamentals

After completing this lesson, you will be able to:

- understand web development and differentiate between a website and a web application.
- create a static website using HTML/CSS in an appropriate environment
- create dynamic websites using JavaScript as the frontend scripting
- implement common algorithms that use sequence, selection, and repetition in JavaScript
- determine ways of debugging their code in JavaScript



3.1 Introduction

The World Wide Web's (WWW) development has made it easier to share information and data in many formats, which is becoming more prevalent and essential in our everyday lives. Documents, picture, audio, video are the main types of information but not limited to. Especially, with the rise of social media, the links and tags of information enable you to share and search related information on the internet. Additionally, the dynamic nature allows changing the contents for individuals and group of peoples. For example, the contents of the website remain same whoever visits it, but after logging in as a member customized data is shown. Hyper-Text Markup Language (HTML) is the primary language that is used for the basic website development. Though different software and tools allow you to create a website using a template, but basic knowledge of HTML is necessary to customize it according to your desire.

3.1.1 Web and Website

A document which exists and is accessible through internet is a webpage, while a set of webpages is a termed as **Website**. For example a news website Associated Press of Pakistan as shown in figure 1, has different sections and each one of them has at least one webpage in every section. To access a webpage, software namely **web browser** is used. You just provided in the Universal Resource Locator (URL) in the website which is the generally accessible address of the document. This way, web browser will locate the document and display it.

Fig-1: Website of Pakistan's Official News Agency

In case, you are unable to recall the URL of the website you are searching, help of search engines is quite useful. **Search engine** provides the service to seek relevant information based on the keywords you have entered. The search engine on the basis of keywords creates different combinations and searches the relevant information. Apart from the relevant matches, auxiliary results are also displayed to help user to explore extra information and viewpoints. Once, the results are collected by the search engine, it displays the website address and little content from every website; in a list fashion as shown in fig. 2. This way, it is easier to select the website, you are looking for.

The point to note is that every website has a 'Homepage'. As soon as a webpage opens up, its homepage is displayed and thereafter you can navigate the website to extract information. Every website and relevant pages need to be uploaded to a web hosting services, which offers web servers that are available round the clock. This way, clients and visitors can access the website on their digital devices.

National Book Foundation

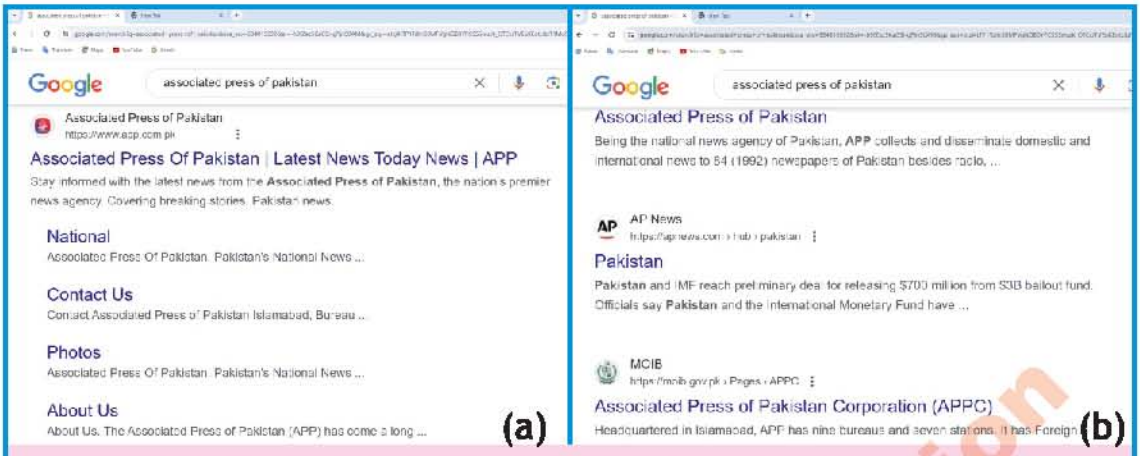


Fig.2: Search Results in a search engine (a) Primary result (b) Aux. results

3.1.2 Web Application

On the other hand, a computer program which offers a service or executes tasks via a browser and internet connection, remotely accessing a server, is called web application. As shown in fig.3, through web browser, a web application can be visited by users like a Customer Relations Management (CRM) system which handles retailing, supplies, promotions, customer feedback, etc. At the backend there can be more than one server, for each type of task or a single server handling all the requests and managing it accordingly, while keeping the front-end updated. It all depends on the architecture which is being deployed to facilitate the front-end, and transparent to a normal user.

3.1.3 Website Development

Website is the first step to show your presence in the digital world. Website only shares information and contents and does not allow any changes by the viewer. A website can have single or multiple pages linked together. For example, you can create a personal page of your interest highlighting your hobbies, activities and passion. You can share your ambitions and achievements. Like, if photography is your hobby, you can upload picture albums on the website on a separate page. Anyone who visits the website can view the contents but in no manner can manipulate it.

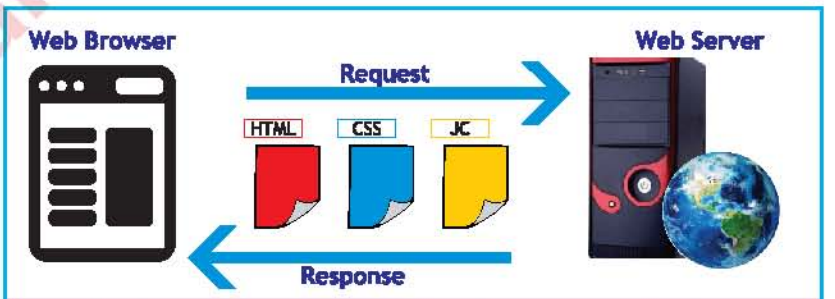


Fig-3: Communication between Web Server and Web Browser.

3.1.4 Static Website

Regardless, how much content and how many pages you increase in your website as discussed above, the contents remain unchanged; unless you change it yourself. In other

National Book Foundation

words the website is static and once accessed by the user; it will be loaded from the server where you have hosted it. After loading, on the user's computer the link to the server is no more required. Such static websites are easy to create and load on the client's site.

3.1.5 Dynamic Website

A website is dynamic if the information is changed or adjusted in accordance with user input or choice. For example, alteration of background color of a webpage every time a user clicks a button. So, why a website needs to be dynamic? The intention of every website owner is to have frequent visitors and the visitors should stay long. For example, online shopping sites offer promotions and packages. So that the visitors lengthen their stay on the interested page, get awareness and shop. To achieve this, dynamicity in websites is applied using scripts, like JavaScript, python, PHP, ASP, Net, etc. In the context of online-shopping site, every member of the website can customize its webpage based on his/her interest as shown in fig. 4, while non-members and visitors get the same page to view, every time.

Fig-4: Features of a dynamic website of a e-store

3.1.6 Front-End Development

A front-end of a website provides the interface which is graphical nowadays and termed as Graphical User Interface (GUI). The person visiting your website views and interacts with GUI. The front-end of a website is developed using Html, CSS and Javascript, etc. A person, who develops such front-end websites and GUIs, is termed as 'Front-end Developer'.

3.1.7 Back-End Development

Front-end developed websites and GUIs need to communicate with the server for every event which the user generates and corresponding result is to be displayed on the front-end. This bridging between the front-end and the server is taken care by back-end development. A person who writes code about such services that are provided by the website is called a 'Back-end Developer'. Back-end development requires more knowledge and skill level in hand than front-end development, like knowledge of JavaScript, Python, PHP, ASP, Net, etc.

National Book Foundation

3.2 HTML

Hypertext Markup Language (HTML) is the language used to define and display your contents in the form of a webpage. With the help of tags, you will define different contents what they are, correspondingly HTML will display them accordingly. Html identifies and provides support for every object in a webpage on the basis of tags. For example, "This is my first attempt for a webpage." is a sentence which you want to get displayed in a web browser, so you have to put it like this:

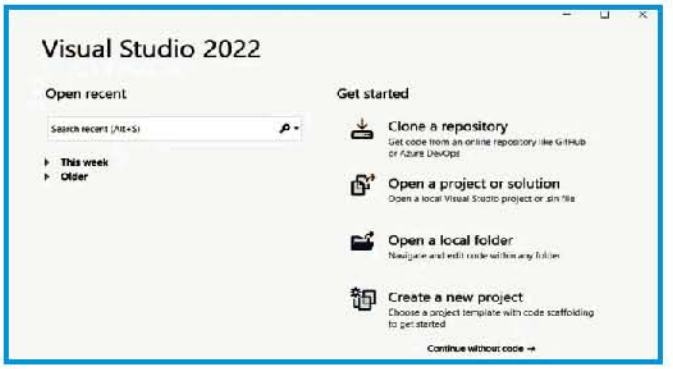
"This is my first attempt for a webpage."

Character(s) between angle brackets '<' and '>' are called tags. The said character(s) is as per the defined HTML rules and is one of the elements that we can use. Every component of Html is identified by a starting and terminating tag. Additionally, we have to define from where to start and end the effect of every component. Therefore, we place , the starting marker, in the start of the sentence and , the closing marker, after the sentence. Now, you have put your sentence between two tags with a component to take effect. The 'p' tag is used for paragraph/sentences.

Similarly, there are tags for everything in HTML. But, the main point to note here is that you can write your html code in a notepad and run it in a browser. There are many softwares/IDE environments available that help in minimizing your coding efforts. Visual Studio, Netbeans, etc. For this chapter, we will be using the Visual Studio version 2022 environment, but the codes mentioned here will be applicable to any other environment, as well.

Installing Visual Studio

For installing the latest Integrated Development Environment (IDE) of Microsoft Visual Studio, visit https://visualstudio.microsoft.com/vs/ and download the installer. The installer is an executable file (.exe extension) and you just need to double click to start the installation.



Microsoft Visual Studio is a wide-ranging IDE which can be used for writing and running code for more than 30 languages. Therefore, it allows you to choose the language environment. Once selected, click "install" and it will download and install the environment on your system (PC/ Laptop). After the installation, you can run the IDE from your system.

National Book Foundation

3.2.1 HTML Document Object Model

The Document Object Model (DOM) is a standard which provides mutual interpretation where grammar of a language can be associated with and can coexist on various operating systems. In HTML, every file is interpreted as a DOM-tree where hierarchy of the said file is defined. As discussed in last chapter, a tree consists of nodes and links, so is the case here where every object and component of HTML is treated as node. Hence, the file, components, features, script and even comments which exist in a web page all are treated as objects by HTML as shown in fig 5.

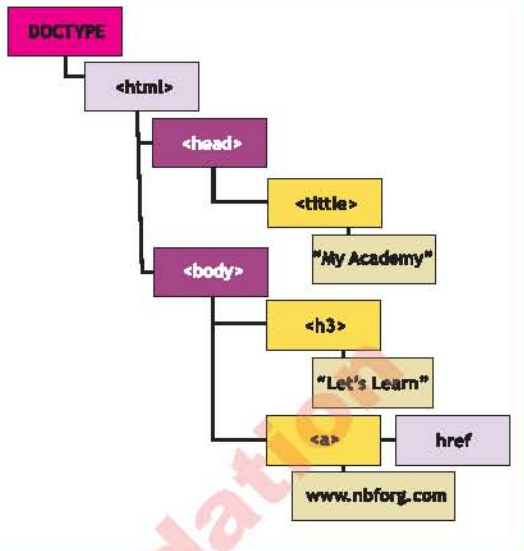


Fig-5: General DOM Tree

Simpler approach in locating any object in a web page via accessing the DOM-tree is 'element-id'. So, we will call function like 'getElementById()' and it will either return the object of the element we seek if it is able to locate it else a null value is returned.

```
1.  <!DOCTYPE html>
2.  <html>
3.  <body>
4.  <p id="leid"> * </p>
5.  <p id="hdom"> $ </p>
6.  <script>
7.  var elem_1 = document.getElementById("leid");
8.  var elem_2 = document.getElementById("hdom");
9.  document.getElementById("leid").innerHTML = "Text from first 'p' is: " +
10. elem_1.innerHTML;
11. document.getElementById("hdom").innerHTML = "Text from second 'p' is: " +
12. elem_2.innerHTML;
13. document.write(elem_1.innerHTML);
14. document.write("<br/>");
15. document.write(elem_2.innerHTML);
16. </script>
17. </body>
18. </html>
```

Output

```
Text from first 'p' is: *

Text from second 'p' is: $

Text from first 'p' is: *
Text from second 'p' is: $
```

Fig-6: Locate and print elements by ID

In fig-6, we will locate elements using tag <p> via defining the IDs, 'leid' (short for Locate Element by ID) and 'hdom' (short for title of this section). Thereafter, on line 6th and 7th two variables are defined which store the located elements. Next, two ways are listed to print the values. 'innerHTML' is used for attaining (and if needed for updating) values of an element.

National Book Foundation

3.2.2 Tags in HTML

However, since html is a tag based programming language, so we should recognize ourselves with the most important and frequent tags that we will encounter in the development of learning curve. With a little bit of practice, these tags will be on your fingertips, as the tags themselves are quite self-explanatory.

3.2.2.1 HTML Tag

Any html is identified with the <html> and </html> tag-pair, i.e. anything written between this tag-pair is recognized as a html document. An HTML-Document is arranged just like a simple document that you write and prepare in Notepad, Wordpad or MS Word, etc., with headings, sub-headings, sentences, etc. Similar to <html> tag-pair, everything is characterized according to its respective tag-pair. The file is saved with extension .html.

3.2.2.2 Head Tag

The <head> tag-pair is defined, where tags can be placed which are not part of the main body of html, like the title of the webpage.

3.2.2.3 Title Tag

Just like every document has a name, the <title> tag-pair allows your webpage a name. This way, even if multiple pages or tabs are open in the browser, a webpage is easily identifiable.

3.2.2.4 Body Tag

The main part of the html is the 'body', all the main constituents of the document are arranged in the <body> tag-pair.

3.2.2.5 Tag for Headings

The headings are of different levels provided by html and there are 6 defined levels, 1 being the biggest and 6 is the smallest heading. The heading tag pair looks like <h2>...</h2>, for a second level heading.

3.2.2.6 Tag for Line Break

To split a sentence into multiple lines a <br/> tag is used. If this tag is not used between 2 sentences, and even you write the second sentence on a new line; html does not recognize this style and will put both the sentences in the same line, one after the other.

The following code in fig 7 shows how you can setup your first web page. The title of the page is between head tag-pair followed by the body, and is shown on the browser's tab in the output. Note that everything written in the body is displayed on the webpage. Thereafter, you can assign suitable headings and paras, with necessary line breaks.

National Book Foundation

```
1.  <html>
2.  <head>
3.     <title>Main Page</title>
4.  </head>
5.     <body>
6.           <h1>Main Heading of the Page</h1>
7.              <p>An introduction about the page is to be put here telling the visitor what the page is all about.<p>
8.                    <h2>Sub-Heading can follow</h2>
9.                    <p>It is a good approach to split the information on your website into various sections.<br/>
10.                   This way, it will be easier to navigate and manage the information on the page.</p>
11.                   <h2>Sub-Headings are not limited</h2>
12.                   <p>Why limit to one only, the more the better. But good approach is to segregate the
13.                   information into groups and assign each group under a different subheading.</p>
14.    </body>
15. </html>
```

Output

Fig-7 Output of various HTML tags

3.2.2.7 Tag The Text

Span is used to provide style and arrangement to a line. For multiple lines you need to assign the tag-pair on every line. Whereas, provides the same effect to a set of lines present in the page. The style and classes, etc. are generally applied in the div tag-pair.

The 'i' tag-pair is used for a sentence to be in italics, like a note. The tag-pair is for emphasis, whereas <b> and tag-pair are used for display of bold and strong characters. Frequently used tags are listed in Table 3-1.

Fig 8 shows result of various text-tags, like "I" tag-pair used in line 20.

Teacher's Guide

W3Schools offers a thorough introduction to HTML which can help students understand the organization of static content by going deeper into the fundamental components of webpages.

(https://www.w3schools.com/html/)

Tag-Pair	For the Task
p	Paragraph, sentence
b	To make characters bold
i	Text is shown in italics
em	When you need to emphasize a word but with italics.
strong	When emphasize a word but with bold
sup	Superscript, helpful in formula and footnotes
sub	Subscript, helpful in formula and footnotes
u	Underline a text
small	Smaller text size, like footnote

Table-3-1: List of Frequent Tags used in Text

National Book Foundation

```
1. <html>
2.   <title>Main Page</title>
3.   <body>
4.          <h2> Main Heading of size 2 </h2>
5.          <h6> Smallest Heading</h6>
6.          <p> We put a simple sentence, here. </p>
7.          <span> Span is Marked in this line.</span>
8.          <span> In this line as well because we didn't assign line-break before this line, so this line is in
9.          continuation.</span>
10.         <br/>
11.         <div style="color    blue;">
12.                 Let's explore 'div' tag along.
13.                 <p> 'div' tag changed color to blue </p>
14.                 <br/>
15.                 <center>
16.                    <h3> A Level 3 Heading in the center</h3>
17.                    <p>Just a sentence with the heading, otherwise, why do we need the heading.<p>
18.                 </center>
19.         </div>
20.         <i>A horizontal line to differentiate sections or mark the end of the text. </i>
21.         <br/>
22.         </em>in italics, as well</em>.
23.         <br/>
24.         <b>Nothing of Document Text follows.</b>
25.         <br/>
26.         <strong>But we can have a Footnote, below the line.</strong>
27.         <hr />
28.         <hr>
29.         <small>But we can have a Footnote, below the line.</small>
30.   </body>
31.</html>
```

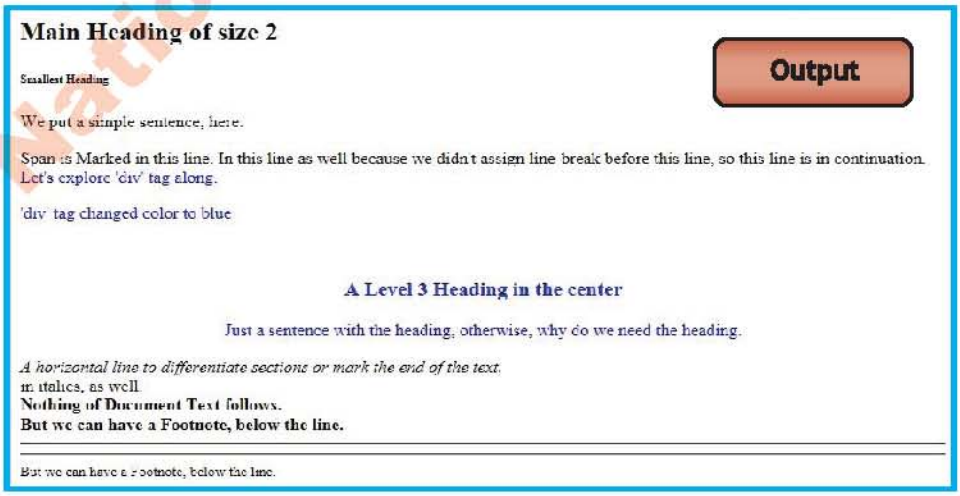


Fig-8: Text Variations using Tag

National Book Foundation

So, as you have observed so far, that HTML provides support for a document to be presentable just like a word-editor. Additionally, we can insert an image in the webpage by assigning the 'src' meaning the path where the file is located and providing the name of the file, as shown in fig 9.

3.2.2.8 Bullets & Numbering

Html treats bullets and numbering in the form of an unordered <ul> and ordered list <ol>, respectively. Bullets can be of type circle, square or disc. The numbered list have the option of numerals and alphabets to choose from, just like a word-editor.

```
1.  <html>
2.   <title>Main Page</title>
3.    <center>
4.            <img src="mat-logo.jpg" alt="MAT-Logo">
5.    </center>
6.   <body>
7.    <center>
8.            <h2> My Teaching Academy </h2>
9.            <h4> Let's Learn </h4>
10.   </center>
11.           <p> Welcome to website of My Teaching Academy. </p>
12.           <br/>
13.           <div style="color   blue;">
14.           <center>
15.           <h3> Subjects We Teach </h3>
16.           <ul>
17.                   <li>English</li>
18.                   <li>Maths</li>
19.                   <li>Urdu</li>
20.                   <li>History</li>
21.                   <li>Computer Science</li>
22.           </ul>
23.           <p>
24.           </center>
25.           <h3> Fee Structure </h3>
26.           </div>
27.    <p>We take fee in advance. <b>Discount</b> is offered on payment of <strong>Full Fee</strong> only.</p>
28.   <br/>
29.           <h4> Option of Installments</h4>
30.           A maximum of 4 installments can be arranged, on monthly basis:<br/>
31.           <ol>
32.                   <li>First Installment is 40%</li>
33.                   <li>Second installment is 20%</li>
34.                   <li>Third Installment is 20%</li>
35.                   <li>Fourth installment is 20%</li>
36.           <ol/>
37.           <hr>
38.           <small>We have No-Refund Policy.</small>
39.   </body>
40. </html>
```

Output

Fig-9: Bullets / Numbering in Text

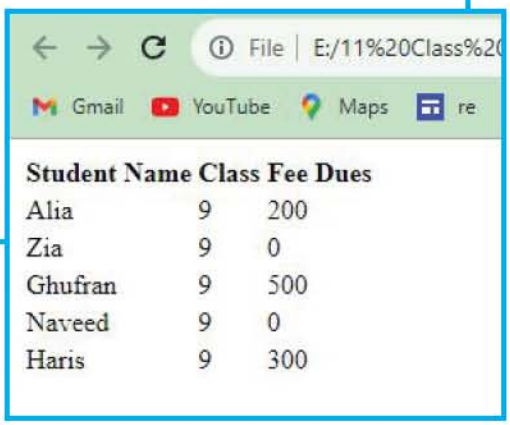
National Book Foundation

3.2.3 Manipulating Data with Tables

Tables in HTML

Tables are a good way to enlist data which is visually appealing. Html provides 'table' tag-pair to allocate and designate data within the table. The first row of the table is the header row and is generally used for headings and is defined with the 'th' tag-pair. Whereas data is manipulated through 'td' tag-pair.

```
1.  <html>
2.    <body>
3.      <table>
4.      <tr>
5.        <th>Student Name</th>
6.        <th>Class</th>
7.        <th>Fee Dues</th>
8.      </tr>
9.      <tr>
10.       <td>Alia</td>
11.       <td>9</td>
12.       <td>200</td>
13.       </tr>
14.     <tr>
15.       <td>Zia</td>
16.       <td>9</td>
17.       <td>0</td>
18.     </tr>
19.     <tr>
20.       <td>Ghufran</td>
21.       <td>9</td>
22.       <td>500</td>
23.     </tr>
24.     <tr>
25.       <td>Naveed</td>
26.       <td>9</td>
27.       <td>0</td>
28.     </tr>
29.     <tr>
30.       <td>Haris</td>
31.       <td>9</td>
32.       <td>300</td>
33.     </tr>
34.     </table>
35.   </body>
36. </html>
```



Output

fig-10: Table of student records with Fee submitted.

National Book Foundation

3.2.4 Links to Resources

Links are helpful components of a webpage, via which you can redirect to another webpage or a document. Links are called Hyperlinks in html with 'a' tag-pair. Hyperlinks are easy to identify on a webpage, as the mouse cursor changes as soon as the cursor touches a link element. Hyperlinks can be associated not only to text, but to images as well. The general syntax for defining a links is like "<a href="url">link text</a>", where href refers to the address alongwith the path and link-text is for user information. Target is an optional parameter but its value defines where to open the webpage. If the value chosen is 'blank', it will open in a new tab or window; however for the value of 'self' the destination address will open in the same tab or window. Fig-11 depicts result of adding the following line of code in the body of the webpage:

```
<a href="https://www.nbf.org/" target="_blank">National Book Foundation </a>
```

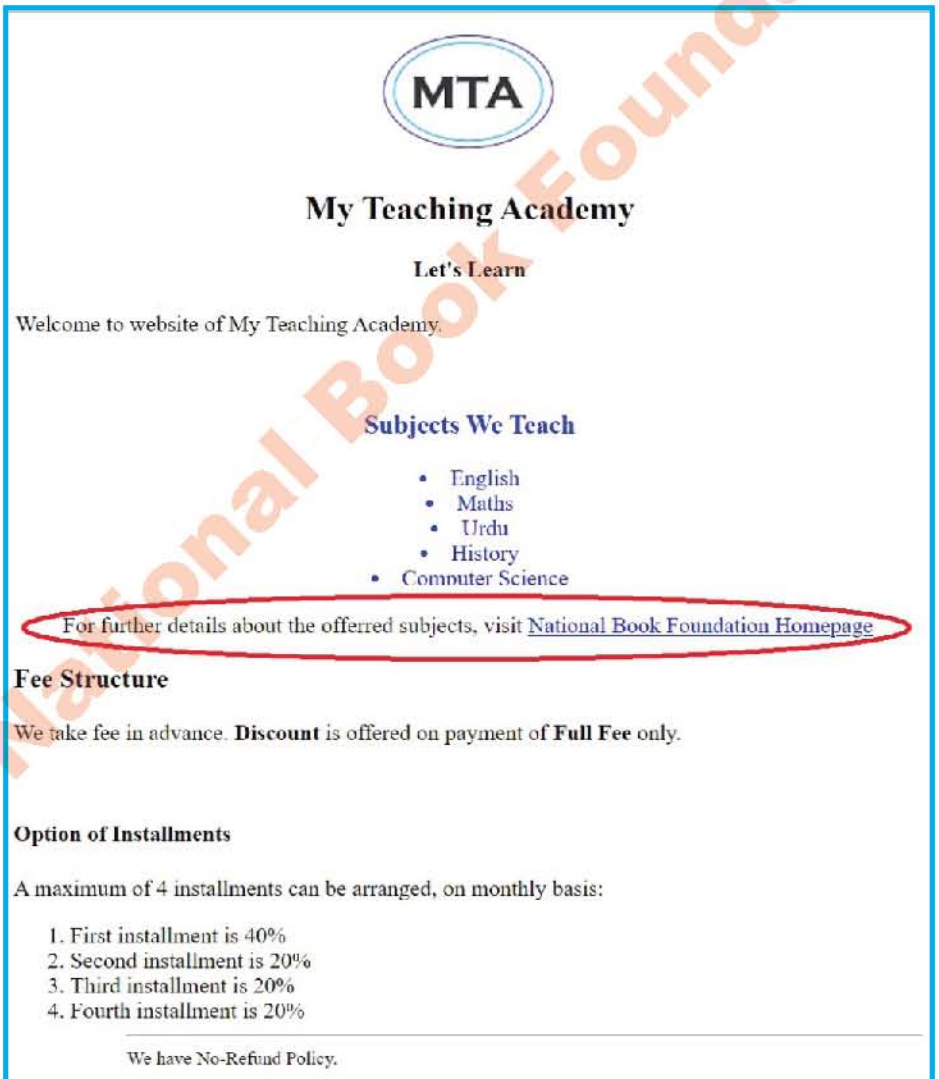


Fig-11: Adding a Resource Link in a webpage

National Book Foundation

3.3 Cascading Style Sheets (CSS)

In the development of HTML webpage the scheme, arrangement and presentation of the whole webpage along with the components are handled by a stylesheet language. There are various stylesheet languages available like Cascading Style Sheets (CSS), Document Style Semantics and Specification Language (DSSSL), Extensible Stylesheet Language (XSL), etc. The most common and frequently used with HTML is CSS. So, this way the contents of the webpage are defined through HTML while the look of different components is handled via CSS. This way, it is easier to manage and troubleshoot your web designing code for extension and digging out the errors.

There are three ways, via which we can use CSS styles in our HTML webpage.

Inline CSS:

Any CSS attribute that we want to incorporate can be added using a HTML tag (like the ones, we have covered in the last section) and incorporated in the body section, as shown in fig-12.

```
1. <p style="color:red; font-size:40px;
2. font-style:italic; text-align:center;">
3. My Teaching Academy
4. </p>
```

Fig-12: Inline CSS sample

Embedded (Internal) CSS

Instead of assigning styles for every heading and other component at the time of its first occurrence in the code, a better approach is to outline all the styles in the header under the tag-pair of 'style' as shown in fig-13. This way, all the presentation related CSS code is separated and do not indulge with the already written HTML code. Additionally, change in one line in the CSS section will be reflected throughout the respective components.

```
1.  <head>
2.  <title>Main Page</title>
3.  <center>
4.  <img src="mat-logo.jpg" alt="mat-
5.          logo.jpg">
6.  </center>
7.    <style>
8.          body {
9.             color:   olive;
10.              }
11.               h2 {
12.            color:    red;
13.            font-size:30px;
14.                font-style:italic;
15.            text-align:center;
16.          }
17.            </style>
18. </head>
```

Fig-13: Embedded CSS sample

National Book Foundation

External CSS

Alternatively, a file with extension '.css' can be made and all relevant CSS code according to your schema can be present there. Once, the contents of HTML are finalized, just attach the CSS file in the head portion of HTML by passing the link. External CSS are used with large projects, like in commercial purposes.

```
<link rel="mystylesheet" href="my_own_SS.css" />
```

NOTE: *The priority of Inline is highest, followed by embedded styles and lastly the attributes of external are considered; if all three are present in a webpage.*

3.3.1 Decorating Tables with CSS

By using CSS, you can provide borders in table as well. For this purpose, we initially need to state the 'style' tag-pair just before the start of table and later inside the block, we need to set which style to opt for and where to apply it. As shown in the following code, as shown in fig 14 we are defining border for table, table header cells and every other cell in the table. Other than that, we can also designate a background color for the table.

```
1. <style>
2.   table, th, td {
3.         border: 1px solid black;
4.   }
5. </style>
```

Student Name	Class	Fee Dues
Alia	9	200
Zia	9	0
Ghufran	9	500
Naveed	9	0
Haris	9	300

Fig-14: Applying Borders to Table

We may add background color of the table as well, by adding the following code, as shown in fig-15 in 'style' tag-pair.

```
1. <style>
2.   th, td {
3.         background-color: aqua;
4.   }
5. </style>
```

Fig-15: Provide table with a background color

National Book Foundation

3.3.2 Homepage Decor

The code of fig-16(a) is similar to that of fig-11.As a first step, we define 'style' tag-pair in the body and add lines 22 to 24, which result in changing the color of 3rd level headings in the body to green as shown in fig-16 (b). In line 4, 'alt' image tag is alternative wording which describes the picture that is being shown on the page, in a short phrase.

```
1.      <html>
2.      <title>Main Page</title>
3.      <center>
4.      <img src="mat-logo.jpg" alt="mat-logo.jpg">
5.      </center>
6.      <body>
7       <center>
8.      <h2> My Teaching Academy </h2>
9.      <h4> Let's Learn </h4>
10.     </center>
11.     <p> Welcome to website of My Teaching Academy. </p>
12.     <br/>
13.     <div style="color:   blue;">
14.     <style>
15.         body {
16.     color:    red;
17.       background-Image: url("openbook.jpg");
18.       background-repeat: no-repeat;
19.       background-attachment: fixed;
20.       text-decoration-line: underline;
21.       text-decoration-style: wavy;}
22.     h3 {
23.     color:    green; }
24.     </style>
25.     <center>
26.     <h3> Subjects We Teach </h3>
27.     <li>English</li>
28.     <li>Maths</li>
29.     <li>Urdu</li>
30.     <li>History</li>
31.     <li>Computer Science</li>
32.     <p>
33.     <br/><br/>
34.     <div style="color:   black;">
35.     For further details about the offerred subjects, visit
36.     <a href="https://www.nbf.org.pk/" target="_blank">National Book Foundation Homepage</a>
37.     </center>
38.     </div>
39.     <h3> Fee Structure </h3>
40.     </div>
41.     <p>We take fee in advance. <b>Discount</b> is offered on payment of <strong>Full Fee</strong> only.</p>
42.     <br/>
43.     <h4> Option of Installments</h4>
44.     A maximum of 4 installments can be arranged, on monthly basis:<br/>
45.     <ol>
46.     <li>First installment is 40%</li>
47.     <li>Second Installment is 20%</li>
48.     <li>Third installment is 20%</li>
49.     <li>Fourth installment is 20%</li>
50.     <ol/>
51.     <hr>
52.     <small>We have No-Refund Policy.</small>
53.     </body>
54.     </html>
.
```

Fig-16(a): Code for Applying CSS Style to headings and body

National Book Foundation

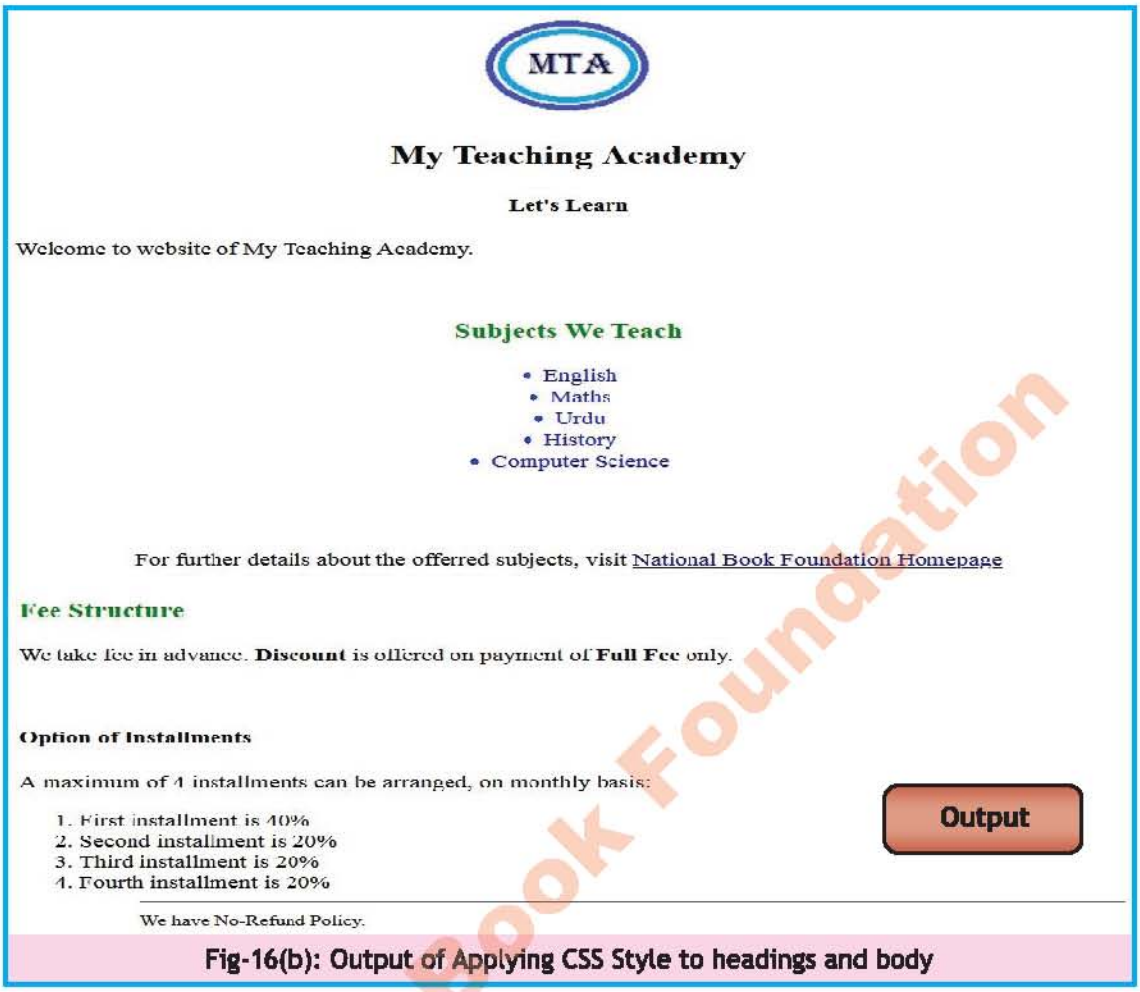


Fig-16(b): Output of Applying CSS Style to headings and body

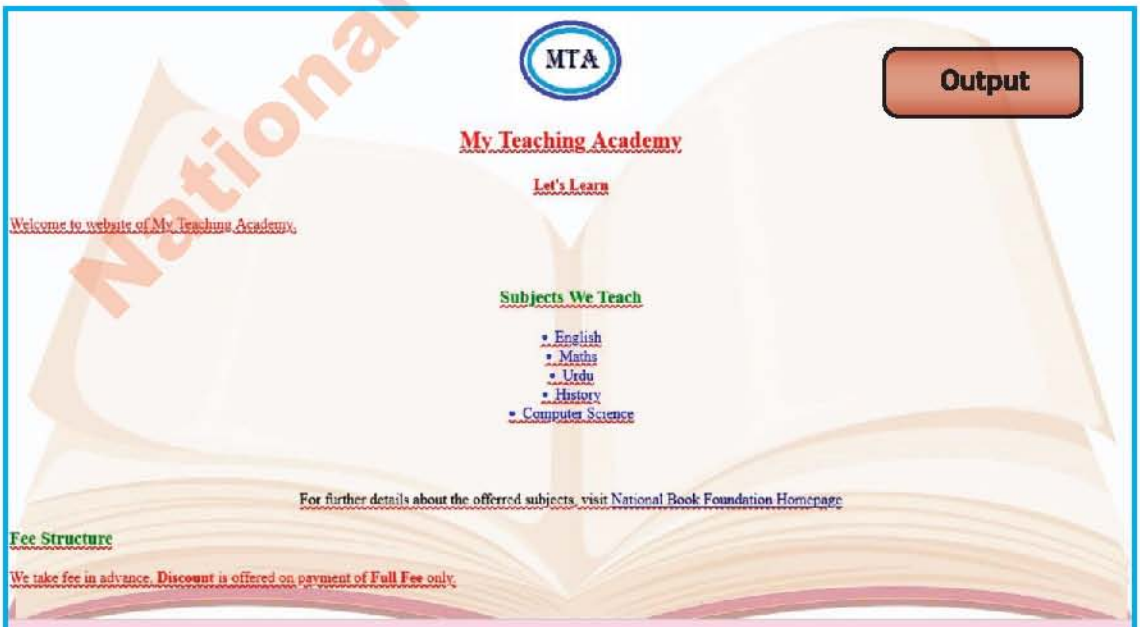


Fig-17: After adding Background Image and Underlined Text in webpage

National Book Foundation

Thereafter, if we add line 14-21, then basically we are applying a new style to the whole body of the webpage. On line 17 of fig. 16(a) we are loading an image of an open book at the background of the webpage and kept it fixed in a non-repetitive fashion in the next couple of lines. In lines 20 and 21, we further extend our décor by underlining the text in the whole page and define the style of underline to be wavy, as shown in fig-17.

File Edit Selection View Go Run ··· Search

<html> Untitled-1

```
1.  <html>
2.  <head>
3.  <style>
4.  body {
5.      overflow: hidden;
6.      margin: 0;
7               padding: 0;
8.      font-size: 2rem;
9.      }
10. #scroll-container {
11.     white-space: nowrap;
12.     position: absolute;
13.     top: 50%;
14.     left: 35%;
15.     animation: scroll 10s linear infinite;
16.     }
17. @keyframes scroll {
18.             0% { transform: translateX(100%); }
19.             100% { transform: translateX(-100%);}
20.                 }
21. .colored-message {
22.         display: inline-block;
23.         animation: linear infinite;
24.         color: purple;
25.           }
26.  </style>
27. </head>
28. <title>Main Page</title>
29. <center>
30. <img src="mat-logo.jpg" alt="mat-logo.jpg">
31. </center>
32. <body>
33. <div id="scroll-container"><span class="colored-message">W e l c o m e </span></div>
34. <center>
35. <h2> My Teaching Academy </h2>
36. <h4> Let's Learn </h4>
37. </center>
38. <p> Welcome to website of My Teaching Academy. </p>
39. <br/>
40. <div style="color: blue;">
41. <style>
42. body {
43. color: red;
44.   background-image: url("openbook.jpg");
45.   background-repeat: no-repeat;
46.   background-attachment: fixed;
47.   text-decoration-line: underline;
48.   text-decoration-style: wavy;}
49. h3 {
50. color: green; }
```

National Book Foundation

```
51.  h2 {
52.    text-shadow: 3px 3px ■black; }
53.  </style>
54.  <center>
55.  <h3> Subjects We Teach </h3>
56.  <li>English</li>
57.  <li>Maths</li>
58.  <li>Urdu</li>
59.  <li>History</li>
60.  <li>Computer Science</li>
61.  <p>
62.  <br/><br/>
63.  <div style="color:■black;">
64.  For further details about the offerred subjects, visit
65.  <a href="https://www.nbf.org.pk/" target="_blank">National Book Foundation Homepage</a>
66.  </center>
67.  </div>
68.  <h3> Fee Structure </h3>
69.  </div>
70.  <p>We take fee in advance. <b>Discount</b> is offered on payment of <strong>Full Fee</strong> only.</p>
71.  <br/>
72.  <h4> Option of Installments</h4>
73.  A maximum of 4 installments can be arranged, on monthly basis:<br/>
74.  <ol>
75.  <li>First installment is 40%</li>
76.  <li>Second installment is 20%</li>
77.  <li>Third installment is 20%</li>
78.  <li>Fourth installment is 20%</li>
79.  <ol/>
80.  <hr>
81.  <small>We have No-Refund Policy.</small>
82.  </body>
83.  </html>
```

Fig-18: Horizontal Scrolled Text on a web page

National Book Foundation

A simple and impressive decor on the website can be a scrolling message. As shown in Fig-18, we added a 'Welcome' message which continuously scrolls horizontally on our web page. We defined a container and assigned the parameters of the text. Next, we defined the horizontal margins where the text should move, using keyframes parameter. Lastly in .colored_message we provided with the color of the text. So far everything is in the 'head' tag-pair, so that as soon as the webpage loads this effect takes place. The remaining portion of the output, i.e. the lower one is same as that of fig-17.

On line 33 inside 'body', using id 'scroll-container', we provided the text to be displayed. In the output, the resultant moving 'Welcome' message is encircled blue.

Activity

For every subject mentioned in Fig-18, create a link which opens in a new tab and assign main contents of the said subject and display the title cover of the relevant book(s).

3.3.3. Adding a Video Clip in Website

To add a video clip in a website, <video> tag is used where you can define the clip size and how the clip should be available at the time when website loads up. Line 7 of the code in fig-19 (a) uses a video tag and specifies width and height of the clip in terms of pixels to adjust the size of the video clip. Additionally, the controls parameter defines that play/pause and volume controls are enabled when the video loads up as shown in fig-19 (b). In the next line, the source and type of the video are specified. Though, Html supports many formats, but among different browsers the most commonly supported video type is MP4.

```
1.  <!DOCTYPE html>
2.  <html>
3.  <body style="text-align: center">
4.      <h2 style="color:    green">Pakistan's First Ever Win in International
5.        Football</h2>
6.      <p>Congratulations !!!</p>
7.      <video width="500px" height="500px" controls>
8.          <source src="fifa-win.mp4" type="video/mp4" />
9.      </video>
10.  </body>
11. </html>
```

Fig-19 (a): Sample code to load a video clip in a website

National Book Foundation

Alternatively, autoplay parameter can be used instead of controls. Autoplay automatically loads the video clip as the webpage loads but does not provide any controls for volume or play/pause. Autoplay can be used by specifying additional features like muted, looped, etc. When muted is used, the clip loads but volume is muted and needs to be unmuted manually while looped will keep the clip running, again and again in a loop, unless it is interrupted manually. Options for manual intervention are accessible by right-clicking the mouse. Fig. 19 (c) is the output of following code:

```
<video width="500px" height="500px" autoplay muted>
```

Fig-19: Loading of a video clip with (b) control parameter (c) autoplay parameter

National Book Foundation

Tip

A good programming practice is the provision of additional video formats like Ogg in video tag-pair too, like:

```
<source src="fifa-win.mp4" type="video/mp4" />
```

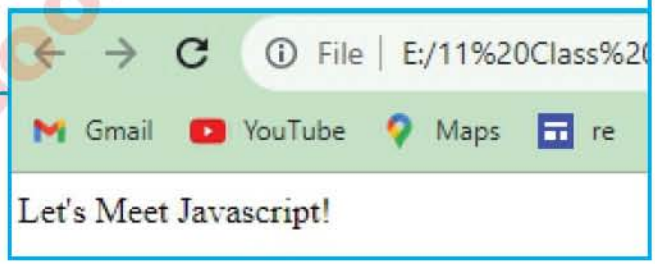
3.4 Java Script

Javascript is an exciting language primarily used in development of web pages and scripts. It does not consume much of memory and that is why it is used at the client-end in developing websites, for making pages dynamic. It easily works with programming languages like Java and HTML, on any operating system.

3.4.1 Let's Meet Javascript

Javascript code can be embedded in HTML with starting and ending tag of <script>, in a webpage. There is no limitation of where to place the code inside a HTML file. For example, the following Javascript code embedded in the body, displays a sentence (string) using 'document.write()' function, as shown in fig-20.

```
1.  <html>
2.   <body>
3.    <script language="javascript" type="text/javascript">
4.      document.write ("Let's Meet Javascript!")
5.  //   document.write ("     Let's Meet Javascript     !          ")
6.    </script>
7.   </body>
8.  </html>
```



Output

Fig-20: A simple JavaScript Text is displayed.

In the sentence, displayed above we put additional spaces and tabs, which javascript simply discarded. For example, we will get the same output if the statement is like the one commented on line 5. Try it for yourself.

In programming everything that a user or another program does with your program that can be sensed and triggers some task to be done, is called an 'event'. Events are important and critical in the functioning and flow-control of your program.

Tip

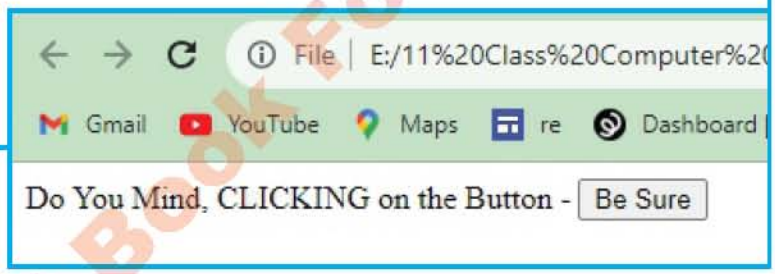
Tip: Javascript like C/C++, uses '//' for line comments and '/* .. */' for block comments.

Similarly, in website development, easiest way to introduce dynamicity is to allow some event to occur and respond accordingly. Alert is the commonly used functionality that

National Book Foundation

Javascript provides to inform user, about the result of his action or notification. Event based code like 'onclick' are put between the start and end of <head> tag; unlike the above example, where we put our code within the <body> tag. The scenario in fig-21, allows the message to be displayed when the button is clicked.

```
1.  <html>
2.   <head>
3.    <script type="text/javascript">
4.        function msgSure() {
5.              alert("Are You Sure???")
6.        }
7.    </script>
8.   </head>
9.   <body>
10.     Do You Mind, CLICKING on the Button -
11.     <input type="button" onclick="msgSure()" value="Be Sure" />
12.  </body>
13. </html>
```



Output

Fig-21: Adding an event via a button.

And after the 'Be Sure' button is pressed, the generated alert is as follows.

You can try for yourself, by replacing different messages and even arithmetic operations like:
alert(2+2)
which results as:

National Book Foundation

In the following line of code in fig-22, two variables are defined. One is a boy's name in the variable 'name' while other is 'reward' having value 5000. So, it displays the result in browser.

```
1.  <html>
2.   <body>
3.    <script type="text/javascript">
4.
5.          var name = "Ahsan";
6.
7.          // document.write(reward);
8.
9.          var reward;
10.         reward = 5000;
11.
12.         document.write(name, " gets a reward of Rupees ", reward);
13.   </script>
14.  </body>
15. </html>
```

Output

Fig-22: Input from the user was, number: 7

Now, let's amend the same program for user input, where we want to change the value 'reward'; by taking input from the user. For input from the user we can use the prompt() function, which pops up a message window like that of alert() function. This input will be assigned to a variable at the backend which can be used, later. A sample code is as follows.

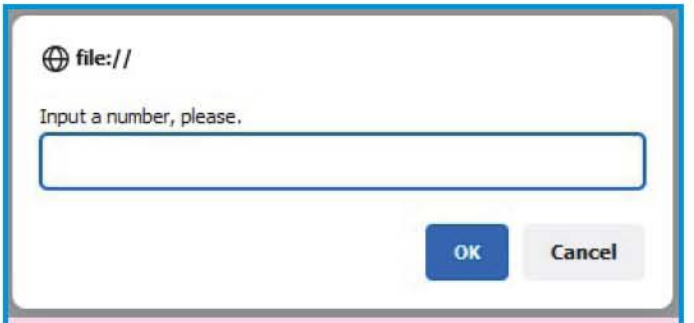


Fig-23: An Alert Window for User Input.

```
1.  var ip = prompt("Input a number, please.");
2.  document.write("Input from the user was, number :", ip)
```

You can replace these lines in the previous example and will get the outputs, as shown in fig-23.

National Book Foundation

3.4.2 Variables

A variable is an entity that stores some value for later use. In mathematics, a variable is generally represented by a single character, but not limited to. Similarly, in programming languages, variable should be named in a meaningful manner. Additionally, the type of values the said variable can store is another important aspect. Such that the developer after defining and assigning a variable, is later able to recall about the task and type of values the said variable holds. The basic value that a variable can hold in JavaScript is either a number or set of characters (called string) or a Boolean which is either 'true' or 'false'. It is critical to note that variable naming convention does not support a number to be the first character of the variable name. Since, Javascript is case sensitive; therefore 'reward' and 'Reward' are two different variables. Table 3-2 enlists basic datatypes of JavaScript.

	Sample Value
Number	7 3 5 9
String	J A V A
Boolean	TRUE

Table3-2: Basic datatypes supported by Javascript with sample values.

Teacher's Guide

Experienced users may choose feature-rich code editors like Visual Studio Code, but beginners can start with simple text editors like Notepad. Real-time collaboration is made possible by online tools such as JSFiddle (https://jsfiddle.net/).

A variable is declared with the 'var' keyword and multiple variables can be declared in the same line of code, too. The first ever assignment of a value to a variable in the life span of program is called 'initialization'. A good programming practice is to declare and initialize the variable at the same time.

```
1.  <html>
2.  <body>
3.     <script type="text/javascript">
4.           var a,b,reward;
5.           reward = 5555;
6.           a = reward % 10;
7.           b = reward / 10;
8.           document.write(a, " — ",b);
9.     </script>
10. </body>
11. </html>
```

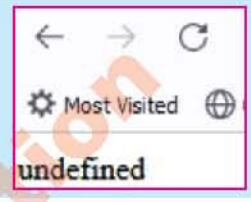
Output

Fig-24: Multiple variables.

National Book Foundation

It is important to note that sequence of instructions in programming matters. For example let's look at the following code, where we defined two variables and keeping the good practice in mind, declared and initialized variable name in line 7. However, just to highlight how things can be alternatively done, we declared the reward variable in line 8 but initial value as assigned to it in the next line. In line 12 we print the statement using both the variables, as shown in fig-24.

Note that, if you take line 7 and put after line 11, then the following message is displayed.



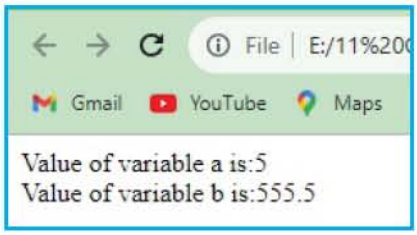
This is due to the reason that the variable is defined after the write statement is using the variable. Unless, the variable is declared and holds a value, only then it can be used as per the programming sequence. Hence, sequence of instructions is important and though you might have written a program syntax-wise correctly, but the change of sequence might lead you to unexpected bugs.

3.4.3 Operators

Javascript supports arithmetic operators to be used which are Addition (+), Subtraction (-), Multiplication (*) and Division (/). Other than this, the Modulus (%) operator can also be used which gives remainder of a division operation.

The code shown in fig-25, takes 3 variables namely, 'a', 'b' and reward. Only reward is initialized through a constant value in line 5. In line 6 and 7, remainder of reward divided by 10 is stored in 'a' and 'b' is assigned the value when reward is divided by 10, respectively.

```
1.  <html>
2.  <body>
3.    <script type="text/javascript">
4.          var a,b,reward;
5.          reward = 5555;
6.          a = reward % 10;
7.          b = reward / 10;
8.          document.write('Value of variable a is:',a);
9.          document.write("<br />")
10.         document.write('Value of variable b is:',b);
11.   </script>
12. </body>
13. </html>
```



Output

Fig-25: Handling and Printing multiple variables.

National Book Foundation

3.4.4 Conditional Statement

Conditional or selection statement is an essential part of the program where amongst choices, the program chooses on the basis of some constraint. Applying an 'if' statement before one or more lines of code on the basis of some condition is met makes a typical selection scenario. That is, if the condition is met, then those line(s) will be executed otherwise skipped.

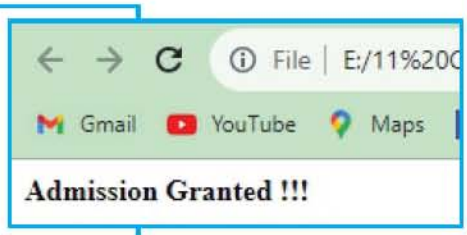
Now, to check the condition, Javascript provides set of comparison operators to be used for evaluating the condition. The conditional operators are listed in table.

Operator	Name	Example
==	Is equal	x == y
!=	Is not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Table:3.3: JavaScript Conditional Operators

For example, the admission office of a Montessori school checks the age of a child, if the kid is of at least 4 years old, then admission is granted. So, the code should look something like, as shown in fig-26.

```
1.  <html>
2.   <body>
3.    <script type="text/javascript">
4.          var kid_age = 5;
5.          if( kid_age > 3 )
6.          {
7.             document.write("<b> Admission Granted !!! </b>");
8.          }
9.    </script>
10.  </body>
11. </html>
```



Output

Fig-26: 'if' statement.

You may check it for different values of the variable and also for different comparison operators.

A better notion is to align both the scenarios, i.e. if condition is met and vice versa. This is achieved using an 'if-else' statement, as shown in Fig-27. This way, either of the two situations will definitely happen.

National Book Foundation

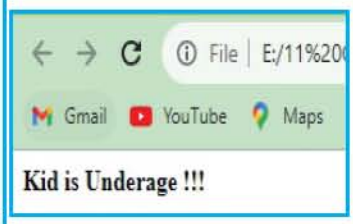
```
1.  <html>
2.    <body>
3.      <script type="text/javascript">
4.            var kid_age = 3;
5.            if( kid_age > 3 ){
6.              document.write("<b> Admission Granted !!! </b>");
7.            }
8.            else {
9.              document.write("<b> Age Requirement is not met !!! </b>");
10.           }
11.     </script>
12.   </body>
13. </html>
```

Output

Fig-27: 'if-else if' statement.

There are scenarios where more than two possibilities exist and for that reason, we can modify our selection statement to be an 'if-else if-else' statement. This way, the set of conditions apply, first with 'if' and thereafter with 'else if' statements. For any other condition that has not been catered for, 'else' will take care of it. In Fig-28 the previous code is extended by checking multiple conditions.

```
1.  <html>
2.    <body>
3.      <script type="text/javascript">
4.      var kid_age = 3;
5.      if( kid_age > 3 ){
6.         document.write("<b> Admission Granted in Nursery Class !!! </b>");
7.      }
8.      else if (kid_age > 4){
9.         document.write("<b> Admission Granted in KG Class !!! </b>");
10.     }
11.     else if (kid_age > 6){
12.        document.write("<b> Admission Should be taken in Primary School !!!
13. </b>");
14.     }
15.     else {
16.        document.write("<b>  Kid is Underage !!! </b>");
17.     }
18.     </script>
19.   </body>
20. </html>
```



Output

Fig-28: 'if-else if-else' statement.

Activity

Take a number as input from the user and write a program to calculate the grade of the student in a particular subject. If the number is greater than equal to 90 then assign A+ grade, for 80 and above but below 90, A-grade is designated. Similarly in the range of 70-79, B grade needs to be awarded. If the number lies between 60-69, C grade should be printed. In case the number is between 50 and 59, D grade and less than 50, F grade should be displayed on screen.

National Book Foundation

3.4.5 Iterative Statement

Iterative statement like 'For Loop' is used to get similar kind of task done. Rather than writing the same line of code multiple times, the same task is achieved in much lesser line of code. The 'for loop' works on the basis of an index, which you can initialize in the loop. Next is the terminating condition which needs to be set for the loop to terminate. Lastly, step-size needs to be defined that how many steps the index will take after each iteration; till the terminating condition is met. In the following example 'for loop', an index is initialized to 0, the value of index will increment with 1 and loop will execute till index value remains less than 10. Fig-29 depicts start, end and iteration of a for-loop.

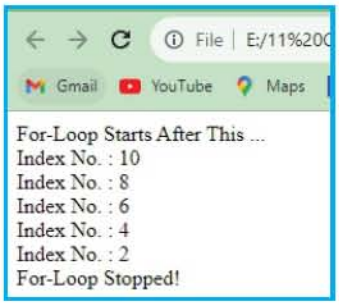
```
1.  <html>
2.    <body>
3.      <script type="text/javascript">
4.              var index;
5.              document.write("For-Loop Starts After This ... <br />");
6.              for(index = 0; index < 10; index=index+1){
7.                      document.write("Index No. : ", index, "<br />");
8.              }
9.              document.write("For-Loop Stopped!");
10.     </script>
11.   </body>
12. </html>
```

Output

Fig-29: A simple For-Loop, shows how index value increases.

Alternatively, we can decrease the index value and set the condition accordingly; in the code of fig-30 we increased the step size, too.

```
1.  <html>
2.    <body>
3.      <script type="text/javascript">
4.              var index;
5.              document.write("For-Loop Starts After This ... <br />");
6.              for(index = 10; index > 0; index=index -2){
7.                      document.write("Index No. : ", index, "<br />");
8.              }
9.              document.write("For-Loop Stopped!");
10.     </script>
11.   </body>
12. </html>
```



Output

Fig-30: For-Loop with step-size of 2.

National Book Foundation

3.4.6 Nested Loops

Multiple iterative tasks, if can be related then they can be incorporated in such a way that one loop can reside inside the other and are termed as 'Nested Loop'. In nested loop, initially the outer loop will start and then the inner loop will run and finish. Thereafter, the index value of outer loop will increment and the inner loop will start and end again, and so forth and so on; till the outer loop terminates. Fig-31 demonstrates a nested loop.

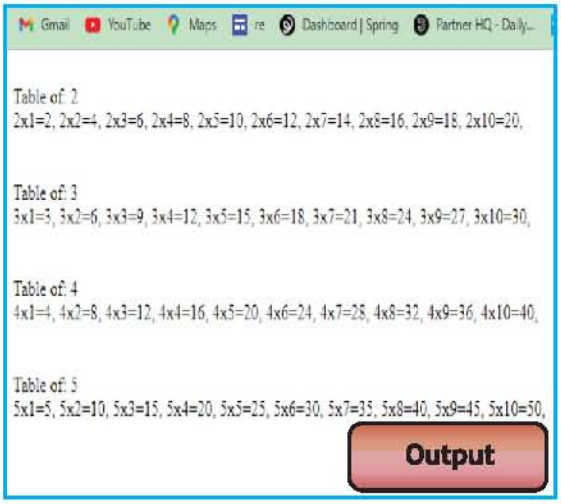
```
1.  <html>
2.   <body>
3.    <script type="text/javascript">
4.        document.write("About to Enter the Loops !!!<br /> ");
5.           for (var i = 0; i < 5; i++)
6.            {
7.                 document.write("<br /> The Outer Loop: ",i,"<br />");
8.                 for (var j = 0; j < 3; j++)
9.                 {
10.                    document.write("Inner Loop: ",j,", ");
11.                }
12.       }
13.       document.write("<br /> <br /> <br /> Loops Terminate !!!<br /> ");
14.    </script>
15.   </body>
16. </html>
```

Output

Fig-31: Nested Loop

In a similar fashion, the code in fig-32 prints Mathematical Tables from 2 to 5. The outer loop assigns the value, for which the table is required while the inner loop prints the table.

```
1.  <html>
2.   <body>
3.    <script type="text/javascript">
4.       for (var i = 2; i <= 5; i++)
5.       {
6.          document.write("<br /> Table of: ",i,"<br />");
7.          for (var j = 1; j <= 10; j++)
8.          {
9.             document.write(i,"x",j,"=",i*j,", ");
10.         }
11.      document.write("<br /> <br /> ");
12.      }
13.
14.   </script>
15.  </body>
16. </html>
```



Output

Fig-32: Printing Mathematical Tables from 2-5.

National Book Foundation

3.4.7 Arrays

An array is a datatype which can hold a number of homogenous set of elements. Such that we do not need to define multiple variables of the same type like num1, num2, ... num25. Instead, we can declare an array which contains 25 values. This way, we can directly access any value just by passing the respective array-index number. It is trivial to note that array-index starts with 0. Declaration of an array is of the form as shown in line 7 of fig-33.

```
1.  <html>
2.    <head>
3.      <title>Arrays in JavaScript</title>
4.    </head>
5.    <body>
6.      <script type="text/javascript">
7.              var Arr = [4, 2, 5, 1, 3];
8.              document.write("Array is:", Arr);
9.      </script>
10.   </body>
11. </html>
```

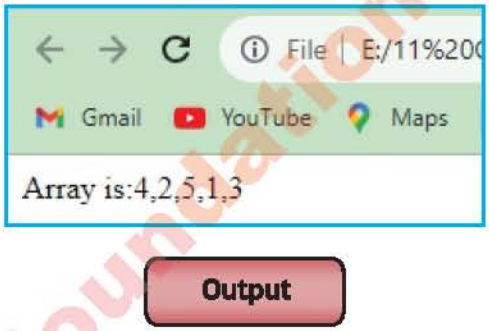


Fig-33: Array declaration and initialization in a single line.

Alternatively, we can declare a null array first and assign values to it later, as shown below:

```
1.  <html>
2.    <head>
3.      <title>Arrays in JavaScript</title>
4.    </head>
5.    <body>
6.      <script type="text/javascript">
7.              var Arr = []
8.              Arr[0] = 4;
9.              Arr[1] = 2;
10.             Arr[2] = 5;
11.             Arr[3] = 1;
12.             Arr[4] = 3;
13.             document.write("Elements of Array are, after declaring a null array: ", Arr);
14.     </script>
15.   </body>
16. </html>
```

Fig-34: Initialization of an array, element by element.

So, rather than assigning values one by one, as highlighted in the lines 8-12 in above code, we can alternatively use for-loop to populate an array. The code in fig-35 initializes a null-array on line 7. In the next couple of lines a for-loop is taken into account in which user input is taken, via prompt() function.

National Book Foundation

```
1.  <html>
2.   <head>
3.            <title>Arrays in JavaScript</title>
4.   </head>
5.   <body>
6.    <script type="text/javascript">
7.            var Arr = []
8.            for (i =0; i< 5; i++)
9.                    Arr [i] = prompt("Input Array element");
10.           document.write("Elements of Array using For-Loop are, after declaring a null array: ", Arr);
11.   </script>
12.  </body>
13. </html>
```

Output

Fig-35: Use input for multiple values

Once, the user input is taken and the loop terminates, in line 12 the elements of the array are displayed as in pervious example.

Elements of Array using For-Loop are, after declaring a null array: 5,3,7,1,9

3.4.8 Functions

Function is a set of line which occurs in the code quite often that can be segmented once, and called again and again. This way, rewriting the same set of code for similar results can be eliminated. Through functions, different sets of code can be separated resulting in fewer lines of code assigning them meaningful names based on their functionality. This leads to efficiently managing a large computer program. Programming languages provide built-in such functions like earlier we have used functions prompt(), input(), etc. Whenever the function is called, the caller does not necessarily need to know the code behind that function, to use it.

A function has a name through which it is identified and called. Additionally, a function can have arguments which are variables local to that function and their life span is limited to the said function. Variables outside functions are global variables and can be accessed anywhere from the program. Recall the earlier program of fig-11 where we defined a function, namely msgSure() and whenever the button is pressed an alert pops up asking "Are you sure?". In line 4 of the said code, we used the 'function' reserved word to define and named it. However, on line 11, we called the same function just with its name.

National Book Foundation

We extend the same code, and provide it with values which are assigned to arguments of the function. And thus, we can use them in the function, as shown in Fig-36.

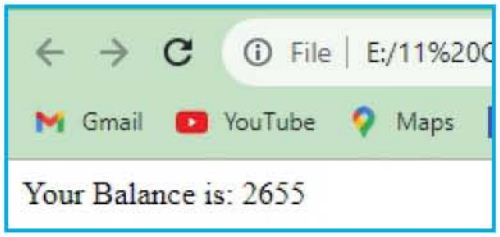
```
1.  <html>
2.   <head>
3.    <script type="text/javascript">
4.          function msgSure(currency, amount) {
5.                  alert("Are You Sure, you gave me "+currency+"."+amount,"???")
6.          }
7.    </script>
8.   </head>
9.   <body>
10.         Do You Mind, CLICKING on the Button -
11.         <input type="button" onclick="msgSure('Rupees', 5000)" value="Be Sure" />
12.  </body>
13. </html>
```

Output

Fig-36: A function with arguments

Lastly, lets define another function calcBill() which is similar to the above as it accepts 2 arguments, namely bill and 'amount_rcvd'. The function subtracts bill from 'amount_rcvd' and returns the result. An important point to note in fig-37 is the lines 5 and 10, where the former prints inside the function and latter displays the result outside the scope of the function. Similar approach can be used for testing the values of a function, when frequent occurring set of codes are selected to form a new function.

```
1.  <html>
2.   <body>
3.    <script type="text/javascript">
4.          function calcBill(bill, amount_rcvd) {
5.                  document.write("Your Balance is: ");
6.                  return(amount_rcvd - bill);
7.          }
8.
9.          var balance =calcBill(2345, 5000);
10.         document.write(balance);
11.   </script>
12.  </body>
13. </html>
```



Output

Fig-37: Function returns value.

National Book Foundation

3.5 Debug The Code

Debugging refers to locating an error or a bug in the code. In Visual Studio .Net, Select Run from the menu and apply choose 'Run Debugging'. Thereafter, you will get the Debugging menu with some buttons and a debug console. Rather than every time accessing the menu and choose options from there to debug the code, the debugging menu is quite handy and allows you to 'Pause/Continue' your debugging code with the first button.

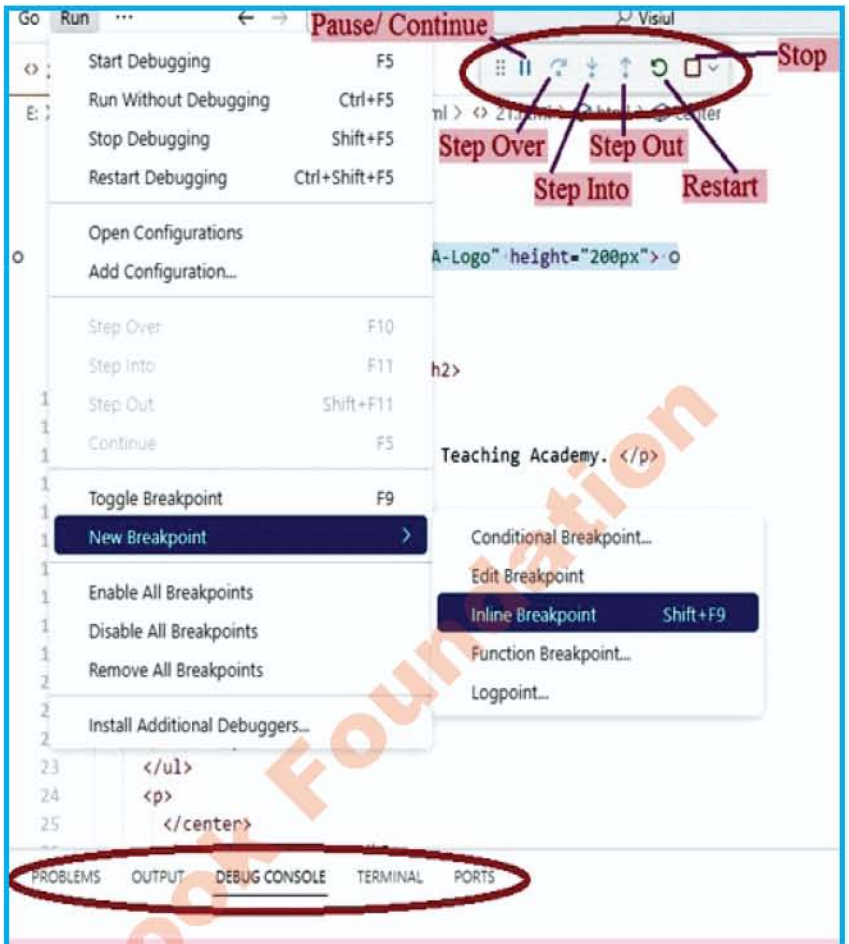


Fig-38: Debugging Menu and Console

'Step Into' will go through the code line by line, as the program will normally execute, so that you have a thorough tracing capability of the code.

To emphasize on lines where you want to check the values of variables, you can assign a breakpoint as shown in fig-38.

You can assign multiple breakpoints for the sake of traceability. In Visual Studio .Net, a breakpoint is shown as a red dot at the start of line, as shown in fig-39. While the code is executed in debug mode, the execution halts at every breakpoint such that behavior of code according to the values can be analyzed. This way, it becomes easier to trace a bug and fix it accordingly.

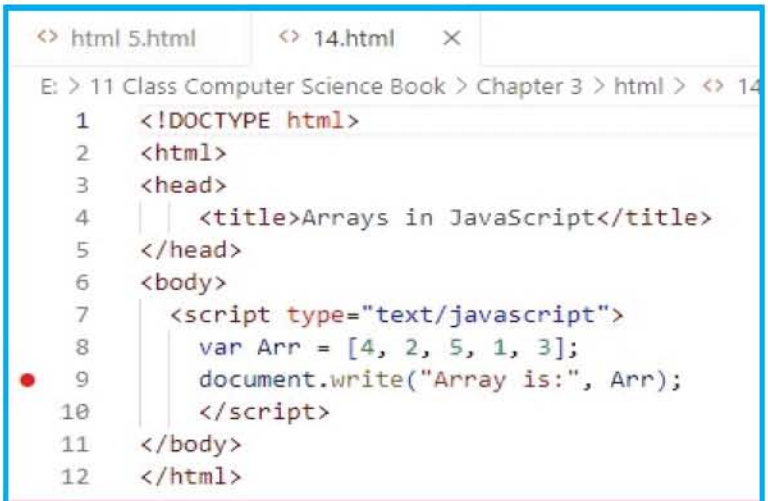


Fig -39 : Assigned Breakpoint in code

National Book Foundation

3.6 Create a Dynamic Website

If a school wants to display the result of each student on their website, then they have to create a different static webpage for each student. And as the students increase, so does the handling and managing of these webpages. The solution is to provide dynamic website such that contents change according to the user input. Such dynamic website can be created via JavaScript along with Html and CSS.

Recall Fig 29, where we printed index number via a loop from 0-9. We extend the same code and the revised code is shown in fig.40 (a), where the initial 9 lines are the same. At line 10, a new function is added namely 'descOrder()' which when is called prints the index in descending order. Additionally, the program has an id of 'dynamic content' and a button which when pressed calls the descOrder() function. A dynamic content is component/ portion of the website that alters based on user input.

```
1.  <html>
2.    <body>
3.      <script type="text/javascript">
4.              var index;
5.              document.write("For-Loop Starts After This ... <br />");
6.              for(index = 0; index < 10; index=index+1){
7.                      document.write("Index No. : ", index, "<br />");
8.              }
9.              document.write("For-Loop Stopped!");
10.             function descOrder () {
11.                     for(index = 10; index > 0; index=index -1){
12.                             document.write("Index No. : ", index, "<br />");
13.             }
14.             }
15.     </script>
16.     <p id="dynamicContent">Click the button to change output to Descending Order.</p>
17.     <button onclick="descOrder()">Descending Order</button>
18.   </body>
19. </html>
```

Fig-40 (a): Code to create dynamic websit

As the webpage loads, the index numbers are printed in ascending order and button with a message is visible as shown in fig- 40(b). On pressing the button, the index numbers are printed in reverse order, as shown in fig- 40(c).

Moreover, after line 13 in the code, if we add the following line of code:

```
document.body.style.backgroundColor = "peachpuff";
```

Save the file, refresh the webpage and press the button, then the output changes to fig-40(d) where the background color of the webpage also changes.

You may try to add another button and separate the functionality of changing the background color.

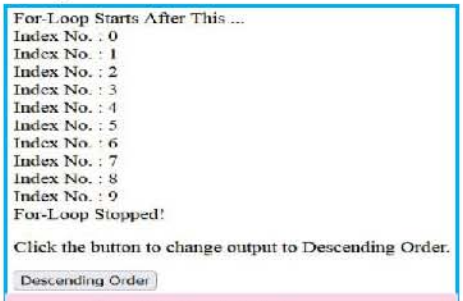


Fig 40 (b): Initial screen

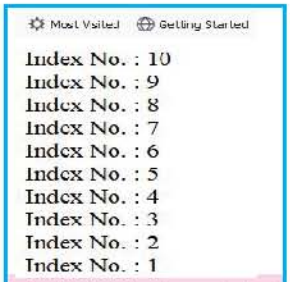


Fig 40 (c): Index printed in Descending Order

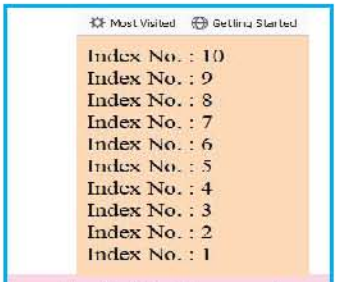


Fig 40 (d): Change of background color dynamically

National Book Foundation

Summary

- A document which exists and is accessible through internet is a webpage, while a set of webpages is termed as Website.
- Universal Resource Locator (URL) is the accessible address of the document.
- Search engine provides the service to seek relevant information based on the keywords you have entered.
- A computer program which executes tasks via a browser and internet connection, remotely accessing a server, is called web application.
- Static Website once loaded on the user's computer the link to the server is no more required. Whereas dynamic website contains pages which are created on demand basis, at the spot.
- The front-end of a website is developed using Html, CSS and Javascript, etc., while back-end development requires more knowledge and skill level than front-end development, like knowledge of JavaScript, Python, PHP, ASP.Net, etc.
- Hypertext Markup Language (HTML) is the language used to define and display your contents in the form of a webpage.
- Anything between angle brackets '<' and '>' and the character(s) between these angle brackets is said to be a tag.
- Links are called Hyperlinks in html with 'a' tag-pair. Hyperlinks are easy to identify on a webpage, as the mouse cursor changes as soon as the cursor touches a link element.
- Arrangement and presentation of the whole webpage along with the components are handled by a stylesheet language, like CSS.
- There are three ways, via which we can use CSS styles in our HTML webpage, i.e. Inline, Embedded or External CSS.
- JavaScript is primarily used in development of web pages and scripts. It does not consume much of memory and that is why it is used at the client-end in developing websites, for making pages dynamic. It easily works with programming languages like HTML, similar to CSS.
- For input from the user we can use the prompt() function, which pops up a message window like that of alert() function.
- A variable is an entity that stores some value for later use. In programming languages, variable should be named in meaningful manner. A good programming practice is to declare and initialize the variable at the same time.
- In conditional statements, on the basis of some constraint the program flow is chosen.
- The 'for loop' starts with an initial value of index, and iterates till the

National Book Foundation

- terminating condition is met; and adjusts every iteration on the basis of step-size.
- Multiple iterative tasks, which are incorporated one loop inside the other and are termed as Nested Loop.
- An array is a datatype which can hold a number of homogenous set of elements.
- Function is a set of code that occurs in the code quite often which can be segmented once and called again and again. A function has a name through which it is identified and called. A function can return a value.

Exercise

Teacher's Guide

The Odin Project's "Debugging Challenges" offers helpful activities specifically designed for JavaScript debugging.
(https://www.theodinproject.com/paths/full-stack-javascript/courses/javascript)

Select the suitable answer for the foll

1) Everything in HTML is identified on the basis of _____.
   a) Brackets b) Title
   c) Tags d) Image
2) The output of HTML code is visible in _____.
   a) Notepad b) File
   c) Browser d) Spreadsheet
3) Name of a web page can be given using _____ tag.
   a) Body b) Title
   c) Head d) Footer
4) Main parts of a document are arranged in _____ tag.
   a) Head b) Title
   c) Body d) Line Break
5) The heading tag-pair for 5th level heading is _____.
   a) <h5 .. h5> b) <h5 .. /h5>
   c) h5> .. </h5 d) <h5> .. </h5>
6) Span is used to provide _____ to a line.
   a) Font b) Border
   c) Style d) Color1)
7) First row of table in HTML is called _____.

National Book Foundation

a) Title row b) Top Row
c) Header Row d) Upper Row'

8) p' tag-pair is used for _____.
a) Print b) Page align
c) Page break d) Paragraph

9) A variable cannot start with a _____.
a) Alphabet b) Number
c) Character d) String

10) The first value assigned to a variable after declaration is called _____.
a) Beginning value b) Starting value
c) Initialization d) Substitution

Give Short answers to the following Short Response Questions (SRQs).

1) Contrast between website and web application.
2) What is 'href' refers to and how to use it?
3) Enlist the optional parameters to open a webpage.
4) List out the frequent tags used in text of a webpage and what are they used for?
5) Explain the role of <body> tag-pair in a document.
6) How the event based code is used in JavaScript?
7) Infer about the External CSS? Where are External CSS generally used?

Give Long answers to the following Extended Response Questions (ERQs).

1) What is Document Object Model? Explain with the help of an example.
2) Write code to differentiate between different types of headings in HTML.
3) Elaborate steps and provide code to load a background image in a webpage.
4) With the help of sample code, highlight different methods to incorporate CSS code in a HTML webpage.
5) Sketch steps and provide code to apply border and color to a table in a webpage.
6) Discuss the functionality JavaScript can provide in a webpage with the help of a suitable example code.
7) Articulate steps and write code to create a scrolling text on a webpage.
8) Enlist steps to add a video clip in a website which starts playing as the web page loads.
9) Cite steps on compiling the result of your last examination in a tabular form and display it in a webpage.
10) In context of Fig. 40(d), add another button namely 'Revert' which when is pressed, it will reverse both the color and index values.

National Book Foundation

Project - 1

Design a webpage for Annual Function of your school.

Provide with:

- Schedule of Events
- List of Invited Guest Speakers
- Major awards that will be distributed by Chief Guest.

Add some relevant text paragraphs and photographs for each point, along with an introductory paragraph.

Project - 2

Create a webpage about you and your hobbies (like stamp collecting, pets, etc.)
The webpage initially should highlight about your family background, your personal achievements and activities. Later, provide details about your every hobby for creating awareness and motivation for others.
You may add some related links for others who want to start and opt for the same hobbies.

Project - 3

Home-made food items are getting popularity, but majority are lacking a website to market their products and new innovations. Your target is to come up with a colorful webpage emphasizing the awareness of organic and home-based meals.
Moreover, you need to put in a menu of food items with the prices and number of persons servings.
The major ingredients need to be listed against each item of the menu with a picture.
Design a corresponding logo for the website, too. And use the same logo as a background for the website.
Lastly, provide contact details and delivery options.

Activity

Create a timetable for the subjects mentioned in the fig-16, and display on a new page by providing a 'Time Table' link on the main page.

Activity

Using '\' and '/' draw a big 'X', as shown:

National Book Foundation

National Book Foundation

Approved by National Curriculum Council, Secretariat
Ministry of Federal Education & Professional Training
vide letter No. F.1-1 (2024)-NCC/DEA/Dir/English, Dated: 04th March 2024

قومی ترانہ

پاک سَرزمین شاد باد! کشورِ حسین شاد باد!
تو نشانِ عزمِ عالی شان ارضِ پاکستان
مرکزِ یقین شاد باد!

پاک سرزمین کا نظام قوّتِ اُخوّتِ عوام
قوم، مُلک، سلطنت پائندہ تابندہ باد!
شاد باد منزلِ مُراد!

پرچمِ سِتارہ و ہلال رہبرِ ترقی و کمال
ترجُمانِ ماضی، شانِ حال جانِ اِستقبال
سایۂ خُدائے ذُوالجلال!

National Book Foundation

National Book Foundation
as
Federal Textbook Board
Islamabad