IT APPLICATIONS FOR MANAGEMENT

M.B.A. First Year

Semester – I, Paper-VI

Lesson Writers

Dr. Neelima Guntupalli
Assistant Professor, Dept. of CS & E,
University College of Sciences
Acharya Nagarjuna University

Dr. D.V. Chandrashekar
Prof. & Head,
PG Dept. of Computer,
TJPS College,,Guntur.

Dr. T. Srinivasa Ravi Kiran
Associate Professor,
Dept. of Computer Science
P.B. Siddhartha College, Vijayawada.

Dr. B. Basaveswara Rao
System Analyst,
Dept. of Computer Science
Acharya Nagarjuna University

Editor:
Prof. R. Sivarama Prasad
Dean, Faculty of Commerce &
Management Studies,
Acharya Nagarjuna University

Director, I/c
Prof. V.VENKATESWARLU
MA., M.P.S., M.S.W., M.Phil., Ph.D.

CENTRE FOR DISTANCE EDUCATION
ACHARAYANAGARJUNAUNIVERSITY
NAGARJUNANAGAR – 522510
Ph:0863-2346222,2346208,
0863-2346259(Study Material)
Website: www.anucde.info
e-mail:anucdedirector@gmail.com

M.B.A. – IT Applications for Management

First Edition 2025

No. of Copies :

Published by:
Prof. V.VENKATESWARLU,
Director I/C
Centre for Distance Education,
Acharya Nagarjuna University

Printed at:

FOREWORD

Since its establishment in 1976, Acharya Nagarjuna University has been forging ahead in the path of progress and dynamism, offering a variety of courses and research contributions. I am extremely happy that by gaining 'A^{+}' grade from the NAAC in the year 2024, Acharya Nagarjuna University is offering educational opportunities at the UG, PG levels apart from research degrees to students from over 221 affiliated colleges spread over the two districts of Guntur and Prakasam.

The University has also started the Centre for Distance Education in 2003-04 with the aim of taking higher education to the doorstep of all the sectors of the society. The centre will be a great help to those who cannot join in colleges, those who cannot afford the exorbitant fees as regular students, and even to housewives desirous of pursuing higher studies. Acharya Nagarjuna University has started offering B.Sc., B.A., B.B.A., and B.Com courses at the Degree level and M.A., M.Com., M.Sc., M.B.A., and L.L.M., courses at the PG level from the academic year 2003-2004 onwards.

To facilitate easier understanding by students studying through the distance mode, these self-instruction materials have been prepared by eminent and experienced teachers. The lessons have been drafted with great care and expertise in the stipulated time by these teachers. Constructive ideas and scholarly suggestions are welcome from students and teachers involved respectively. Such ideas will be incorporated for the greater efficacy of this distance mode of education. For clarification of doubts and feedback, weekly classes and contact classes will be arranged at the UG and PG levels respectively.

It is my aim that students getting higher education through the Centre for Distance Education should improve their qualification, have better employment opportunities and in turn be part of country's progress. It is my fond desire that in the years to come, the Centre for Distance Education will go from strength to strength in the form of new courses and by catering to larger number of people. My congratulations to all the Directors, Academic Coordinators, Editors and Lesson-writers of the Centre who have helped in these endeavors.

Prof. K. Gangadhara Rao
M.Tech., Ph.D.,
Vice-Chancellor I/c
Acharya Nagarjuna University

M.B.A. – Syllabus

SEMESTER-I

106EM24 : IT Applications for Management

COURSE OUTCOMES:

On successful completion of the course the learner will be able to:

- To provide an understanding of fundamentals of computer and networks
- To provide the basics of the processes associated with developing computer based systems for modern organizations.
- To provide a critical understanding on programming basics using C language
- To provide a strong foundation on documentation, presentation, spreadsheets using MS Office.
- To provide basic fundamentals of cyber security issues

Unit- I:

Introduction to Computers: Evolution & Generations of Computers-Elements of computer-Characteristics of a computer-Classification of Computers-Basic Computer Architecture- Input Output devices- Storage devices-Hardware and software-Networks-Types of Networks LAN WAN MAN , Network topologies

Unit-II:

MS Word- Creation of Document-format document-Text Editing and saving-Organising information with tables and outlines- Mail merge -Index Printing **MS PowerPoint** Features of PowerPoint-Creation of slides- Use of templates and slide designs, Slide master, Animation Timings, Action buttons, Rehearse Narration, Slideshow

Unit- III:

MS Excel: Creating and Editing Worksheets-Cell formatting -Creating and using formulas and functions- Use of Macros-Sorting and Filtering data -Working with Graphs and Charts Tables & Pivot Table

Unit-IV:

Introduction to the C Language: Data Types, Variables, Constants, Input/Output, Operators (Arithmetic, relational, logical, bitwise) Selection Statements (making decisions) if and switch statements, Loops-while, for, do-while statements, Functions: Functions basics, user-defined functions, recursive functions, Arrays: Basic concepts, one-dimensional arrays, two-dimensional arrays multidimensional arrays.

Unit-V:

Security and Ethical challenges in IT - Need for Security-Security Threats and Attacks, Malicious Software, Hacking -Security Mechanisms- Cryptography, Digital signature, Firewall Types of Firewalls identification & Authentication Biometric Techniques-Security policies -Need for legislation, cyber laws, cyber security issues

REFERENCE BOOKS:

I. Introduction to computers Peter Norton-Sixth Edition-Tata MC Graw Hill ,2009
2. Introduction to Information Technology. Rajaraman- Prentice Hall lndia,2008
3. Cox et al - 2007 Microsoft Office System step-by-step, first edition,PHI,2007
4. Winston-Microsoft Office Excel 2007 Data Analysis and Business Modeling, first edition, prentice hall India, 2007.
5. Fundamentals of information Technology Alexis Leon Mathews Leon, Vikas Publishing House
6. Computer Applications in Management Ritendra Goel, DN Kakkar New age International publishers
7. Programming Using the C Language: Hutchison, Robert C Mac Graw Hill
8. C Programming, E Balaguruswamy, Tata MC Graw Hill.
9. Let us C, Yaswanth, P.Kanetkar, BPB Publishers

CODE: 106EM24

M.Sc DEGREE EXAMINATION
First Semester
M.B.A.::Paper VI – IT Applications for Management

MODEL QUESTION PAPER

Time : Three hours **Maximum : 70 marks**

Section –A **5X3=15 M**

Answer Any FIVE of the following

1. a) Elements of Computers
   b) LAN
   c) Editing and Saving
   d) Features of Power Point
   e) Cell Formatting
   f) Use of Macros
   g) Data Types
   h) One –Dimensional Arrays
   i) Hacking
   j) Cyber Laws

Section –B **5X8=40 M**

Answer the following questions

2. a) What are the characteristics and classification of computers-Explain

   (OR)

   b) Discuss various types of Networks and network topologies.

3. a) How to creation of Document –format and Index printing.

   (OR)

   b) Explain the use of templates and slide designs.

4. a) How to creating Worksheets and cell formation?

   (OR)

   b) Discuss about use of Macros –sorting and filtering data.

5. a) Explain about Data variables and selection statements.

   (OR)

   b) Explain basic Functions and user-defined functions.

6. a) What is Firewall? Explain about different types of Firewalls identification.

   (OR)

   b) Discuss about Authentication Biometric Techniques

Section –C **1X15=15 M**
(Compulsory)

7. Case Study

Regardless of how high you rate your writing abilities, it's always a worthy idea to check out an expertly written Case Study example, especially when you're dealing with a sophisticated Management topic. This is precisely the case when WowEssays.com collection of sample Case Studies on Management will prove handy. Whether you need to think up an original and meaningful Management Case Study topic or examine the paper's structure or formatting peculiarities, our samples will provide you with the necessary material.

Another activity area of our write my paper service is providing practical writing assistance to students working on Management Case Studies. Research help, editing, proofreading, formatting, plagiarism check, or even crafting completely unique model Management papers upon your demand – we can do that all! Place an order and buy a research paper now.

Free Case Study About Management of Benign Prostatic Hyperplasia
The condition usually starts around 30years of age in men, but mainly surfaces with the associated symptoms around age of 50 years (Thomas & Haitham, 2003). It is a condition cannot be cured hence management usually focuses on reducing the associated symptoms. The type of management that is instituted is placed on how severe the associated symptoms are or how the complications has affected the patients' health (Timothy & James, 2008). It is important to let the patient understand what the BPH is all about. It is a non-cancerous enlargement of the prostate.

CONTENTS

S.NO.	LESSON	PAGES
1.	Evolution, Generations and Classification of Computers	1.1 – 1.14
2.	Computer Elements, Characteristics and Architecture	2.1 – 2.14
3.	Hardware, Software, Network Topologies, and Storage Devices	3.1 – 3.14
4.	MS Word: Document Creation, Formatting and Editing	4.1 – 4.14
5.	MS Word - Mail Merge, Tables and Index Printing	5.1 – 5.18
6.	MS Power Point: Slide Creation, Design and Animation	6.1 – 6.13
7.	Presentation Enhancements and Slideshow Features Structure	7.1 – 7.17
8.	MS Excel: Worksheet Creation, Formatting and Functions	8.1 – 8.13
9.	MS-Excel	9.1 – 9.13
10.	Data Types, Variables and Constants	10.1 – 10.9
11.	Operators and Control Flow Statements	11.1– 11.10
12.	Functions and Recursion	12.1 – 12.8
13.	Arrays	13.1 – 13.14
14.	Cybersecurity: Threats, Attacks and Malicious Software	14.1 – 14.12
15.	Hacking, Cryptography and Digital Signatures	15.1 – 15.14
16.	Firewalls, Authentication and Cybersecurity Policies	16.1 – 16.12

LESSON- 1

EVOLUTION, GENERATIONS, AND CLASSIFICATION OF COMPUTERS

OBJECTIVES:

1. Understand the historical development of computers from early tools to modern intelligent systems.
2. Identify and describe the five generations of computers and their key features.
3. Classify computers based on size, purpose, data handling, and architecture.
4. Familiarize with essential computing terms and emerging technology trends.

STRUCTURE:

1.1 Introduction

1.2 Evolution of Computers

- **1.2.1 Early Mechanical Devices (Before the 19th Century)**
- **1.2.2 Concept of Programmable Computing (19th Century)**
- **1.2.3 Electromechanical and Early Digital Computers**
- **1.2.4 Transition to Electronic Computers (1950s-1970s)**
- **1.2.5 Personal Computing Revolution (1970s-1990s)**
- **1.2.6 Internet and Networking Era (1990s-2000s)**
- **1.2.7 Modern and Future Computing (2010s-Present & Beyond)**

1.3 Generations of Computers

- **1.3.1 First Generation (1940-1956) – Vacuum Tubes**
- **1.3.2 Second Generation (1956-1963) – Transistors**
- **1.3.3 Third Generation (1964-1971) – Integrated Circuits (ICs)**
- **1.3.4 Fourth Generation (1971-Present) – Microprocessors**
- **1.3.5 Fifth Generation (Present & Beyond)**
- **1.3.6 Future of Computers**

1.4 Classification of Computers

- **1.4.1 Classification by Size and Processing Power**
- **1.4.2 Classification by Purpose**
- **1.4.3 Classification by Functionality (Data Handling)**
- **1.4.4 Classification by Architecture**

1.5 Summary

1.6 Key Terms

1.7 Self Assessment Questions

1.8 Suggestive Readings

1.1 INTRODUCTION:

The journey of computers has been marked by remarkable transformations—starting from simple manual counting devices to the ultra-fast, intelligent machines we use today. Over the centuries, the evolution of computers has been driven by continuous technological innovations, shifting from mechanical and electromechanical components to microprocessors, artificial intelligence, and quantum computing. This development has significantly impacted all aspects of life including science, industry, education, healthcare, and communication.

Computers are not only categorized by their historical development but also by how they process data, their architecture, and their intended use. From the massive supercomputers used for weather forecasting to the embedded computers in everyday household appliances, the diversity of computing devices highlights their growing importance in the modern digital world. This document presents a comprehensive overview of the evolution of computers, their generations, and various classifications based on size, purpose, data handling, and architecture.

1.2 EVOLUTION OF COMPUTERS:

Computers have evolved over time from simple mechanical devices to powerful, intelligent systems. This evolution can be categorized based on key technological advancements, improvements in computing power, and their applications in various fields.

1.2.1 Early Mechanical Devices (Before the 19th Century):

Abacus (Around 2500 BC)

- One of the earliest computing tools, invented by the Chinese.
- Used for basic arithmetic operations like addition and subtraction.
- Consisted of a wooden frame with rods and beads for counting.

Pascaline (1642) – Blaise Pascal

- One of the first mechanical calculators.
- Could perform addition and subtraction using a set of gears and wheels.
- Used in tax calculations in France.

Leibniz Calculator (1673) – Gottfried Wilhelm Leibniz

- An improved version of Pascal's calculator.
- Could perform multiplication and division.
- Introduced the binary system, which became the foundation of modern computing.

Jacquard Loom (1801) – Joseph-Marie Jacquard

- Introduced the concept of punched cards for automated weaving.
- Laid the foundation for programmable machines.

1.2.2 Concept of Programmable Computing (19th Century):

Charles Babbage's Analytical Engine (1837)

- Often considered the "Father of the Computer," Babbage designed the first general-purpose mechanical computer.
- The Analytical Engine had key components similar to modern computers:
  - Input: Punched cards
  - Processing Unit: "Mill" (like a CPU)

  - Storage: Mechanical memory
  - Output: Printed results
- Though never built in his lifetime, the concept influenced future computing developments.

Ada Lovelace – First Computer Programmer (1843)
- Worked with Charles Babbage on the Analytical Engine.
- Created the first algorithm intended for a machine, making her the world's first computer programmer.

1.2.3 Electromechanical and Early Digital Computers (20th Century – 1930s-1940s):
Electromechanical Computers
- Used electrical switches and mechanical parts for calculations.
- Examples:
  - Harvard Mark I (1944) – Used for military calculations in World War II.

Development of Binary Computing
- Claude Shannon (1937) – Introduced the idea of using binary (0s and 1s) for digital circuits, forming the basis of modern computing.
- Alan Turing (1936) – Developed the concept of a "Turing Machine," which defined how computers process data.

First Digital Computers
- Colossus (1943-1944) – Built by the British to break German codes in World War II.
- ENIAC (1945-1946) – First general-purpose digital computer, used vacuum tubes for processing.

1.2.4 Transition to Electronic Computers (1950s-1970s):
From Vacuum Tubes to Transistors (1950s-1960s)
- Vacuum tubes made computers large, slow, and prone to overheating.
- The invention of transistors (1947, Bell Labs) led to:
  - Smaller and faster computers.
  - Lower power consumption.
  - More reliability in calculations.

Birth of Programming Languages
- Assembly language replaced machine code for easier programming.
- High-level languages like FORTRAN (1957) and COBOL (1959) allowed more complex applications.

Introduction of Operating Systems (1960s-1970s)
- Early computers required manual control; OS made them more user-friendly.
- Examples:
  - IBM OS/360 – First widely used OS for businesses.
  - UNIX (1969) – A foundation for modern OS like Linux and macOS.

1.2.5 Personal Computing Revolution (1970s-1990s):
The Microprocessor Invention (1971)
- Intel 4004 – First commercial microprocessor, allowing entire computers on a single chip.
- Led to personal computers (PCs) becoming widely available.

Early Personal Computers (1970s-1980s)
- Altair 8800 (1975) – First affordable PC, required assembly by users.
- Apple I & II (1976-1977) – Introduced by Steve Jobs and Steve Wozniak, featured a keyboard and monitor.
- IBM PC (1981) – Standardized the personal computing industry.

Graphical User Interface (GUI) (1980s-1990s)
- Xerox PARC developed the first GUI-based system.
- Apple Macintosh (1984) – Popularized GUI with a mouse, making computers easier to use.
- Windows (1985-present) – Became dominant OS with easy-to-use interfaces.

1.2.6 Internet and Networking Era (1990s-2000s):

The Rise of the Internet (1990s)
- ARPANET (1969) – Early internet prototype used for research.
- World Wide Web (1991, Tim Berners-Lee) – Allowed easy browsing of online content.
- Search Engines (1990s) – Google (1998) revolutionized information retrieval.

Wireless and Mobile Computing (2000s-Present)
- Laptops & Wi-Fi (1990s-2000s) – Enabled portable computing.
- Smartphones (2007-Present)
  - iPhone (2007) introduced touch-based smart devices.
  - Android (2008) became widely used globally.

1.2.7 Modern and Future Computing (2010s-Present & Beyond):

Cloud Computing & AI (2010s-Present)
- Cloud Services (AWS, Google Cloud, Microsoft Azure) – Store and process data online.
- Artificial Intelligence (AI) – AI-based systems like Siri, Alexa, and ChatGPT improve automation.

Quantum Computing (Future Technology)
- Uses qubits instead of binary bits for exponentially faster computing.
- IBM, Google, and other tech companies are developing quantum computers for complex problem-solving.

Future Trends
- Brain-Computer Interfaces (BCI) – Connecting computers to the human brain.
- Nanotechnology – Smaller and more powerful processors.
- Autonomous Systems – AI-powered self-driving cars, robotics, and automation.

The evolution of computers has transformed the world, progressing from simple mechanical devices to powerful AI-driven systems. With advancements in quantum computing, AI, and cloud technology, the future of computing promises even greater innovation, making technology an essential part of everyday life.

1.3 GENERATIONS OF COMPUTERS:

Computers have evolved over multiple generations, improving in size, speed, processing capability, and efficiency. Each generation brought significant technological advancements.

1.3.1 First Generation (1940-1956) – Vacuum Tubes

1. Technology Used

- Used vacuum tubes as the primary electronic component for processing and memory.
- These tubes acted as electronic switches to process data.

2. Characteristics

- Extremely large in size (filled entire rooms).
- High power consumption and generated a lot of heat.
- Slow processing speed and limited storage capacity.
- Used machine language (binary code: 0s and 1s) for programming.

3. Storage and Input/Output Devices

- Punched cards and paper tape were used for input.
- Magnetic drums were used for memory storage.

4. Usage

- Mainly used for scientific calculations, military applications, and basic business operations.

5. Examples

- ENIAC (Electronic Numerical Integrator and Computer) – First general-purpose computer.
- UNIVAC (Universal Automatic Computer) – First commercial computer used for business and government applications.

1.3.2 Second Generation (1956-1963) – Transistors

1. Technology Used

- Replaced vacuum tubes with transistors, which were smaller, faster, and more efficient.

2. Characteristics

- Smaller in size compared to first-generation computers.
- Consumed less electricity and generated less heat.
- Improved processing speed and increased reliability.
- Allowed batch processing, where multiple jobs could be executed sequentially.

3. Storage and Input/Output Devices

- Magnetic tape was introduced as primary storage.
- Used punched cards for input.

4. Usage

- Widely used in banking, business, and scientific applications.

5. Examples

- IBM 1401 – Used in business applications.
- IBM 1620 – Used in scientific computing.

1.3.3 Third Generation (1964-1971) – Integrated Circuits (ICs)

1. Technology Used

- Replaced transistors with integrated circuits (ICs), which contained multiple transistors on a single chip.

2. Characteristics

- Smaller, more reliable, and faster than second-generation computers.
- Lower power consumption and less heat generation.
- Allowed multi-programming, meaning multiple tasks could run simultaneously.
- Used higher-level programming languages such as COBOL and FORTRAN.

3. Storage and Input/Output Devices

- Introduced hard disks for storage.
- Used keyboards for input and monitors for output.

4. Usage

- Used in business, government organizations, and scientific research.

5. Examples

- IBM System/360 – First family of computers that could run different applications.
- PDP-8 – A popular minicomputer.

1.3.4 Fourth Generation (1971-Present) – Microprocessors

1. Technology Used

- Introduced microprocessors, where all computing functions were integrated onto a single chip.
- Used very large-scale integration (VLSI) and ultra-large-scale integration (ULSI) technology.

2. Characteristics

- Extremely small in size (desktop and laptop computers).
- Very fast processing speed and increased storage capacity.
- Graphical User Interface (GUI) introduced, making computers easier to use.
- Used networking and the internet for communication.

3. Storage and Input/Output Devices

- Hard drives, SSDs (Solid-State Drives) for fast storage.
- Mouse, touchscreens, and improved keyboards for input.
- Monitors with high resolution for better output.

4. Usage

- Used in personal computing, business, education, healthcare, entertainment, and industrial automation.

5. Examples

- Intel 4004 – The first microprocessor.
- IBM PC (Personal Computer) – Popularized personal computing.
- Apple Macintosh – Introduced a user-friendly interface.

1.3.5 Fifth Generation (Present & Beyond) – Artificial Intelligence (AI) and Quantum Computing

1. Technology Used

- Focuses on Artificial Intelligence (AI), machine learning, robotics, and quantum computing.
- Uses nanotechnology and parallel processing for faster performance.
- Quantum computers use qubits instead of binary bits for processing.

2. Characteristics

- Intelligent computers that can learn and make decisions.
- Voice recognition, natural language processing, and automation.
- Uses cloud computing for data storage and processing.
- Development of 5G and high-speed internet to enhance connectivity.

3. Storage and Input/Output Devices

- Cloud storage and edge computing for handling massive data.
- AI-powered voice assistants like Alexa and Siri for user interaction.

4. Usage

- Used in self-driving cars, smart assistants, robotics, and healthcare diagnostics.

5. Examples

- IBM Watson – AI-based computing for businesses.
- Google Assistant, Siri, Alexa – AI-powered virtual assistants.
- Quantum Computers (IBM Quantum, Google Sycamore) – Future computing models.

1.3.6 Future of Computers

- Research is ongoing in biocomputing, quantum computing, and AI-based autonomous systems.
- Future advancements may focus on brain-computer interfaces, DNA computing, and holographic computing.
- Computers are expected to become smarter, faster, and seamlessly integrated into daily life.

The evolution of computers has transformed technology, making devices smaller, faster, and more intelligent. From vacuum tubes to AI-driven quantum computers, each generation has revolutionized industries, improving efficiency and connectivity worldwide.

1.4 CLASSIFICATION OF COMPUTERS:

Computers can be classified based on size, purpose, and processing power into categories such as microcomputers, minicomputers, mainframes, and supercomputers. Each type serves different user needs, ranging from personal use to complex scientific calculations.

1.4.1 Classification by Size and Processing Power

This classification focuses on the physical scale, computational capabilities, and typical user base of different computer types.

A. Supercomputers

- **Core Characteristics**
  - **Extreme Processing Power:** Capable of performing trillions of calculations per second (Teraflops, Petaflops, Exaflops).
  - **Massive Parallelism:** Employ thousands to millions of interconnected processors working simultaneously.
  - **Complex Architectures:** Often utilize specialized interconnects and memory hierarchies for high-speed data transfer.
  - **High Cost:** Extremely expensive to design, build, and maintain.
  - **Large Physical Footprint:** Occupy significant space, often requiring specialized cooling systems.
  - **Specialized Software:** Require software specifically designed for parallel processing.
- **Key Applications**
  - **Scientific Research**
    - Climate modeling and weather forecasting.
    - Astrophysics and cosmology simulations.
    - Nuclear weapons research and simulations.
    - Drug discovery and molecular modeling.
    - Materials science and nanotechnology research.
  - **Engineering**
    - Computational fluid dynamics (CFD) for aircraft and vehicle design.
    - Structural analysis and simulations.
    - Reservoir simulation for oil and gas exploration.

- **Data-Intensive Tasks**
  - Cryptographic analysis and code breaking.
  - Large-scale data mining and analysis.
  - Advanced artificial intelligence and machine learning model training.

- **Examples :**
  - **Fugaku (Japan):** Known for its high performance and energy efficiency, used for various scientific and societal challenges.
  - **IBM Summit (USA):** Focused on AI, machine learning, and scientific discovery.
  - **Sierra (USA):** Used for national security applications and scientific research.
  - **Sunway TaihuLight (China):** Notable for its indigenous processor architecture.

B. Mainframe Computers

- **Core Characteristics**
  - **High Reliability and Security:** Designed for continuous operation and robust data protection.
  - **Large Data Handling Capacity:** Capable of processing and managing massive volumes of data.
  - **High Input/Output (I/O) Throughput:** Efficiently handle a large number of concurrent transactions.
  - **Scalability:** Can be upgraded and expanded to meet growing demands.
  - **Virtualization Capabilities:** Often support multiple virtual operating systems and applications.
  - **Centralized Processing:** Typically manage resources and processing from a central point.
- **Key Applications**
  - **Large-Scale Transaction Processing:**
  - Banking (processing millions of daily transactions).
  - Insurance (managing policyholder data and claims).
  - Retail (managing inventory, sales, and customer data).
  - Credit card processing.
  - **Enterprise Resource Planning (ERP):** Integrating various business functions into a unified system.
  - **Government and Public Sector:** Managing large databases (census, social security).
  - **Airline Reservation Systems.**
- **Examples (with brief context)**
  - **IBM Z series:** A long-standing line of mainframe computers known for their reliability and security.
  - **Unisys ClearPath Dorado:** Another prominent mainframe platform used for critical business applications.

C. Minicomputers (Midrange Computers)

- **Core Characteristics**
  - **Balance of Power and Cost:** Offer significant processing power at a more affordable price than mainframes.
  - **Multi-user Capabilities:** Designed to support tens to hundreds of concurrent users.
  - **Smaller Physical Size:** Occupy less space than mainframes.
  - **Versatile Applications:** Suitable for a range of business, scientific, and industrial uses.

  - **Often Serve as Servers:** Commonly used as departmental servers or application servers.
- **Key Applications**
  - **Small to Medium-Sized Businesses:** Accounting, inventory management, customer relationship management (CRM).
  - **Departmental Computing:** Specific departments within larger organizations (e.g., human resources, finance).
  - **Industrial Process Control:** Monitoring and controlling manufacturing processes.
  - **Scientific and Engineering Laboratories:** Data acquisition and analysis.
  - **Web Servers and Database Servers.**
- **Examples (with brief context)**
  - **DEC PDP series (historical):** Played a significant role in the early days of interactive computing.
  - **IBM AS/400 (now IBM i):** Known for its integrated operating system and business-oriented features.
  - **HP 9000 series (historical):** Used for engineering and scientific applications.

D. Microcomputers (Personal Computers - PCs)

- **Core Characteristics**
  - **Single-User Systems:** Primarily designed for individual use.
  - **Microprocessor as CPU:** Utilize a single integrated circuit as the central processing unit.
  - **Affordable Cost:** Relatively inexpensive and accessible to individuals and small businesses.
  - **Variety of Form Factors:** Available in diverse shapes and sizes to suit different needs.
  - **User-Friendly Interfaces:** Typically feature graphical user interfaces (GUIs).
  - **Wide Range of Software:** Support a vast ecosystem of applications for various purposes.

- **Types of Microcomputers**

  Desktop Computers
  - Designed for stationary use on a desk.
  - Components (tower, monitor, keyboard, mouse) are typically separate.
  - Offer more power and expandability compared to laptops at a similar price point.

  Laptop Computers (Notebooks)
  - Portable computers with integrated screen, keyboard, and trackpad.
  - Designed for mobility and convenience.
  - Trade-off between portability, power, and cost.

  Tablet Computers
  - Portable, thin devices with a touchscreen as the primary interface.
  - Often used for content consumption, basic productivity, and web browsing.
  - May have optional detachable keyboards.

  Smartphones
  - Handheld mobile devices with advanced computing capabilities.
  - Integrate phone, internet, multimedia, and application functionalities.
  - Highly portable and connected.

  Workstations
  - High-performance microcomputers optimized for professional tasks.

- Feature powerful processors, high-end graphics cards, large amounts of RAM and storage.
- Used for graphic design, video editing, CAD/CAM, scientific simulations, and software development.

Wearable Computers

- Computing devices designed to be worn on the body.
- Examples include smartwatches, fitness trackers, smart glasses, and health monitoring devices.
- Often have limited processing power but offer specialized functionalities.

E. Embedded Computers

- **Core Characteristics**
  - **Integrated into Larger Systems:** Designed to perform specific control or monitoring functions within a host device.
  - **Dedicated Purpose:** Optimized for a particular task.
  - **Resource Constraints:** Often have limited processing power, memory, and energy resources.
  - **Real-Time Operation:** Many embedded systems require timely and predictable responses.
  - **Variety of Microcontrollers and Microprocessors:** Utilize different types of processing units depending on the application requirements.

- **Key Applications (with more specific examples)**
  - **Automotive:** Engine control units (ECUs), anti-lock braking systems (ABS), airbags, infotainment systems, navigation systems.
  - **Home Appliances:** Washing machines, refrigerators, microwave ovens, smart thermostats, security systems.
  - **Industrial Machinery:** Robotics, automated manufacturing systems, process controllers.
  - **Medical Devices:** Pacemakers, insulin pumps, patient monitoring systems, diagnostic equipment.
  - **Consumer Electronics:** Digital cameras, gaming consoles, smart TVs, set-top boxes.
  - **Aerospace:** Flight control systems, navigation systems, satellite control.
  - **Internet of Things (IoT) Devices:** Smart sensors, connected home devices, wearable trackers.

1.4.2 Classification by Purpose

This classification distinguishes computers based on the breadth of tasks they are designed to handle.

A. General-Purpose Computers

- **Detailed Characteristics**
  - **Flexibility:** Can execute a wide range of software programs.
  - **Programmability:** Their functionality can be altered by changing the software.
  - **Balance of Resources:** Designed with a balance of processing power, memory, and storage to handle diverse tasks.
  - **Wide Availability of Software:** Benefit from a large ecosystem of commercial and open-source software.

- **Examples:** Most desktops, laptops, tablets, smartphones, and even many servers fall under this category.

B. Special-Purpose Computers

- **Detailed Characteristics**
  - **Dedicated Hardware and Software:** Optimized for a specific task, often with custom-designed hardware.
  - **High Efficiency for the Intended Task:** Perform their specific function very effectively and often faster than general-purpose computers.
  - **Limited or No Programmability for Other Tasks:** Their functionality is typically fixed.
  - **Cost-Effective for the Specific Application:** Can be more economical for high-volume, repetitive tasks.
- **Examples**
  - **Weather Forecasting Systems:** Specialized supercomputers with software optimized for atmospheric modeling.
  - **Air Traffic Control Systems:** High-reliability systems with specialized displays and software for tracking aircraft.
  - **Industrial Robots:** Controlled by embedded computers programmed for specific manufacturing tasks.
  - **Medical Imaging Devices (MRI, CT Scanners):** Integrate specialized hardware and software for image acquisition and processing.
  - **Calculators:** Designed solely for arithmetic operations.
  - **Automated Teller Machines (ATMs):** Dedicated to banking transactions.
  - **Video Game Consoles:** Hardware and operating systems optimized for gaming.
  - **Network Routers and Switches:** Specialized hardware for routing and forwarding network traffic.

1.4.3 Classification by Functionality (Data Handling)

This classification focuses on how computers represent and process information.

A. Analog Computers

- **Detailed Characteristics**
  - **Continuous Data Representation:** Use physical quantities (voltage, current, mechanical motion) to represent data.
  - **Calculations Based on Physical Laws:** Perform computations by manipulating these continuous quantities.
  - **Results as Continuous Values:** Output is typically in the form of graphs, dials, or other continuous representations.
  - **Historically Significant:** Played a crucial role in early scientific and engineering applications.
  - **Limited Precision:** Accuracy is often limited by the precision of the physical components.
- **Examples**
  - **Slide Rules:** Use logarithmic scales to perform multiplication and division.
  - **Mechanical Integrators:** Used in differential analyzers to solve differential equations.
  - **Early Flight Simulators:** Employed analog circuits to model aircraft dynamics.
  - **Electronic Analog Computers (historical):** Used operational amplifiers (op-amps) to perform mathematical operations on analog signals.

B. Digital Computers

- **Detailed Characteristics**
  - **Discrete Data Representation:** Use binary digits (bits: 0 and 1) to represent all types of data (numbers, text, images, sound, video).
  - **Calculations Based on Logic and Arithmetic:** Perform computations using Boolean logic and arithmetic operations on binary data.
  - **Results as Discrete Values:** Output is in the form of digital data that can be displayed, stored, or transmitted.
  - **High Precision and Accuracy:** Can achieve high levels of precision by using more bits to represent data.
  - **Versatile and Programmable:** Can perform a wide range of tasks by executing different software programs.
- **Examples:** All modern general-purpose computers (desktops, laptops, servers, etc.) and most embedded systems are digital computers.

C. Hybrid Computers

- **Detailed Characteristics**
  - **Integration of Analog and Digital Components:** Combine the strengths of both analog and digital computing.
  - **Analog Front-End:** Often use analog circuits to acquire and preprocess continuous data from the real world.
  - **Digital Back-End:** Employ digital circuits for logical operations, data storage, analysis, and control.
  - **Data Conversion:** Require analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) to interface between the analog and digital parts.
- **Key Applications**
  - **Medical Monitoring Systems:** Analog sensors measure vital signs (heart rate, blood pressure), which are then converted to digital data for processing, display, and storage.
  - **Industrial Control Systems:** Analog signals from sensors (temperature, pressure) are processed digitally to control actuators and machinery.
  - **Scientific Simulations:** Some complex simulations benefit from analog components for modeling continuous physical processes and digital components for control and data analysis.
  - **Robotics:** Hybrid systems can be used for sensor data processing and motor control.

1.4.4 Classification by Architecture

This classification focuses on the fundamental organization of the computer's hardware components, particularly memory and the CPU.

A. Von Neumann Architecture

- **Detailed Characteristics**
  - **Single Address Space:** Uses the same memory space for both instructions and data.
  - **Shared Data Bus and Address Bus:** The CPU accesses both instructions and data through the same pathways.
  - **Sequential Execution:** Typically fetches and executes instructions sequentially (though modern implementations use techniques like pipelining to improve performance).
  - **Simpler Control Logic:** Easier to design and implement the control unit.

- **Advantages**
  - **Flexibility:** Programs can easily manipulate data as instructions and vice versa.
  - **Efficient Memory Utilization:** Memory can be allocated dynamically between instructions and data.
- **Disadvantages**
  - **Von Neumann Bottleneck:** The shared bus can limit the speed at which instructions and data can be fetched, potentially slowing down processing.
- **Examples:** Most personal computers, laptops, servers, and mainframes utilize the Von Neumann architecture.

B. Harvard Architecture

- **Detailed Characteristics**
  - **Separate Address Spaces:** Uses distinct memory spaces for instructions and data.
  - **Separate Data Bus and Address Bus:** Allows the CPU to fetch instructions and data simultaneously, increasing throughput.
  - **More Complex Control Logic:** Requires more sophisticated control units to manage the separate memory systems.
- **Advantages**
  - **Faster Instruction and Data Fetching:** Enables parallel access to instructions and data, leading to higher performance for certain tasks.
  - **Improved Performance for Real-Time Systems and DSP:** The ability to fetch the next instruction while processing data is crucial for time-critical applications.
- **Disadvantages**
  - **Less Flexible Memory Allocation:** Memory allocated for instructions cannot be directly used for data, and vice versa.
  - **More Complex Hardware Design.**
- **Examples**
  - **Microcontrollers:** Often use Harvard architecture for efficient execution of control programs.
  - **Digital Signal Processors (DSPs):** Benefit from the parallel fetching of instructions and data required for signal processing algorithms.

1.5 SUMMARY:

The evolution and classification of computers reflect an extraordinary journey of technological advancement—from basic mechanical tools to sophisticated AI-driven and quantum computing systems. Each generation has introduced significant improvements in speed, efficiency, and functionality, revolutionizing industries and everyday life. The classification of computers by size, purpose, data handling, and architecture showcases their versatility and specialization in various fields, from scientific research to household devices. As we move forward, emerging technologies like AI, cloud computing, and brain-computer interfaces promise to further transform the role of computers, making them more intelligent, powerful, and seamlessly integrated into our lives.

1.6 KEY TERMS:

1. Microprocessor, Artificial Intelligence (AI), Quantum Computing, Cloud Computing, Von Neumann Architecture

1.7 REVIEW QUESTIONS:

1. What were some of the earliest mechanical computing devices, and how did they function?
2. Explain the major technological advancement that marked the beginning of the second generation of computers.
3. How did the invention of microprocessors revolutionize the computing industry?
4. Differentiate between general-purpose and special-purpose computers with examples.
5. What are the key differences between Von Neumann and Harvard architectures?

1.8 SUGGESTED READINGS:

1. "Computer Fundamentals" by P.K. Sinha and P. Sinha – A comprehensive book covering the basics and evolution of computers.
2. "Computer Organization and Architecture" by William Stallings – A detailed guide on computer system structures and hardware design principles.
3. IEEE Spectrum & ACM Digital Library – Scholarly articles and updates on emerging computing technologies like AI, quantum computing, and embedded systems.

Dr. G. Neelima

LESSON- 2

COMPUTER ELEMENTS, CHARACTERISTICS AND ARCHITECTURE

OBJECTIVES:

The objectives of the lesson are

- Learn what the main parts of a computer are like input, output, CPU, memory, storage, software, and communication devices.
- Understand how the CPU works and what its parts do (ALU, CU, and Registers).
- Know the difference between main memory and storage, and what they are used for.
- Learn what software does and how it helps the computer run and do tasks.

STRUCTURE:

2.1 Introduction

2.2 Major Computer Elements
- **2.2.1 Input Devices**
- **2.2.2 Output Devices**
- **2.2.3 Central Processing Unit (CPU)**
- **2.2.3 Registers**
- **2.2.4 Memory**
- **2.2.5 Storage Devices**
- **2.2.6 Software**
- **2.2.7 Communication Devices**

2.3 Computer Architecture
- **2.3.1 Components of Computer Architecture**
- **2.3.2 Functional Units and Their Roles**

2.4 System Bus

2.5 Summary

2.6 Key Terms

2.7 Review Questions

2.8 Suggested Readings

2.1 INTRODUCTION:

A computer system is a complex assembly of components that work together to perform tasks, process data, and produce meaningful results. These components include both hardware (the tangible parts of a computer) and software (the instructions and programs that drive operations). Understanding these elements is essential for grasping how computers function and how they interact with users and other systems. This lesson explores the fundamental building blocks of a computer system—such as input and output devices, CPU, memory, storage, and communication devices—as well as the architecture that connects them. It also delves into how the CPU executes instructions through units like the ALU, CU,

and registers, and how the system bus enables efficient communication across components. Through this understanding, learners will gain a clear insight into the inner workings of computer systems and their real-world applications.

2.2 MAJOR COMPUTER ELEMENTS:

The main elements of a computer system include

- **Input Devices** – Used to enter data into the computer.
- **Output Devices** – Display or present the processed data.
- **Central Processing Unit (CPU)** – Performs all processing operations.
- **Memory** – Temporarily or permanently stores data.
- **Storage Devices** – Preserve data for long-term use.
- **Software** – Provides instructions to carry out various operations.

2.2.1 Input Devices

Input devices are hardware components used to enter data and instructions into a computer system. These devices act as the bridge between the user and the computer, allowing communication and control.

Common Input Devices

- **Keyboard**
  - The keyboard is the most commonly used input device.
  - It contains keys for letters, numbers, symbols, and special functions.
  - Used mainly for typing documents, entering data, and giving commands.

- **Mouse**
  - A pointing device that detects two-dimensional motion.
  - It is used to move the cursor, click, drag, and drop items on the screen.
  - Comes in various forms such as optical mouse and wireless mouse.

- **Touchscreen**
  - Combines input and output functions.
  - Allows users to interact directly with what is displayed by touching the screen.
  - Commonly used in smartphones, tablets, and ATM machines.

- **Scanner**
  - Converts physical documents, photographs, or pages into digital format.
  - Used in offices, schools, and graphic design work.
  - Can include flatbed scanners, handheld scanners, or barcode scanners.

- **Microphone**
  - Captures sound and sends it to the computer in digital form.
  - Used for voice recognition, video conferencing, and audio recording.
  - Can be built-in or external.

- **Webcam**
  - Captures video or still images in real time.
  - Often used for video calls, live streaming, and surveillance.

- **Joystick and Game Controller**
  - Used mainly for gaming and simulation environments.
  - Allows control over movement and interaction in virtual environments.

- **Biometric Devices**
  - Identify individuals based on physical traits such as fingerprints, retina scans, or facial recognition.
  - Widely used for security and authentication purposes.

2.2.2 Output Devices

Output devices are hardware components that receive data from the computer and convert it into a human-understandable form. They allow the computer to communicate the results of its operations.

Common Output Devices

- **Monitor (Visual Display Unit)**
  - Displays text, images, and videos generated by the computer.
  - Comes in various types such as LCD, LED, and OLED screens.
  - Resolution and screen size affect clarity and detail.

- **Printer**
  - Produces a physical copy of documents, images, or files from the computer.
  - Types include:
    - **Inkjet Printers** – Ideal for color printing with high quality.
    - **Laser Printers** – Faster and suitable for bulk printing, especially text.
    - **Dot Matrix Printers** – Used for carbon copy printing in industrial settings.

- **Speakers**
  - Convert digital audio signals into sound.
  - Used for listening to music, video sound, system notifications, and audio output during presentations or calls.

- **Headphones/Earphones**
  - Personal audio output devices used for private listening.
  - Often used in multimedia applications and communication.

- **Projector**
  - Displays the computer's output on a large screen or wall.
  - Commonly used in classrooms, meeting rooms, and auditoriums for presentations or lectures.

- **Braille Reader**
  - Special output device for visually impaired users.
  - Converts text into Braille characters that can be read by touch.

- **Plotter**
  - A specialized printer used for printing large-scale designs such as engineering drawings, architectural plans, and maps.

- Uses pens to draw continuous lines rather than dots.

2.2.3 Central Processing Unit (CPU)

The Central Processing Unit (CPU) is one of the most critical elements of a computer system. It is responsible for interpreting and executing most of the commands from the computer's hardware and software. Often called the brain of the computer, it handles all processing tasks, including arithmetic, logical, control, and input/output operations.

The CPU consists of the following major components

1. **Arithmetic Logic Unit (ALU)**
2. **Control Unit (CU)**
3. **Registers**

1. Arithmetic Logic Unit (ALU)

Function:

The ALU is responsible for all mathematical and logical operations. It handles:

- Arithmetic operations like addition, subtraction, multiplication, and division.
- Logical operations such as comparisons (greater than, less than, equal to), AND, OR, and NOT.

Role in CPU:

- Processes numerical data from memory.
- Sends results back to the registers or memory.
- Essential for making decisions and performing calculations required by programs.

2. Control Unit (CU)

Function:

The Control Unit directs and coordinates the activities of all parts of the computer. It:

- Fetches instructions from memory.
- Decodes the instructions to determine what needs to be done.
- Directs other parts of the system (ALU, memory, I/O devices) to execute instructions.

Role in CPU:

- Acts like a traffic controller, managing the flow of data and signals across the system.
- Ensures that instructions are carried out in the correct sequence.
- Controls timing and execution of operations.

3. Registers

Registers are small, high-speed memory units inside the CPU. They temporarily store:

- Data currently being processed.
- Instructions fetched from memory.
- Intermediate results from operations.
- Addresses used to access memory.

Types of Registers:

- **Instruction Register (IR):** Holds the current instruction.
- **Program Counter (PC):** Keeps track of the next instruction's memory address.
- **Accumulator:** Stores intermediate results of calculations.
- **General Purpose Registers:** Temporarily hold data during processing.

Role in CPU:

- Enable fast data access and improve overall performance.
- Act as temporary holding spaces for data during computation.

- Work closely with ALU and CU to execute tasks efficiently.

2.2.4 Memory

Memory stores data and instructions either temporarily or permanently to support the functioning of the computer system.

1. Primary Memory (Main Memory)

- **RAM (Random Access Memory)**

– Temporarily holds data and program instructions during execution. It is volatile and loses data when power is off.

- **ROM (Read Only Memory)**

– Contains permanent instructions, such as the boot process. It is non-volatile and retains data even when power is off.

2. Secondary Memory

Used for long-term storage of data and applications.

- **Hard Disk Drive (HDD)** – Stores large volumes of data on rotating magnetic disks.
- **Solid State Drive (SSD)** – Stores data on flash memory chips with faster access speeds than HDDs.
- **CD/DVD** – Optical discs used for storing media, software, or files.
- **USB Flash Drive** – Portable storage devices that connect via USB ports.

2.2.5 Storage Devices

Storage devices are hardware components used to store data permanently or semi-permanently. Unlike memory (RAM), storage devices retain information even when the computer is turned off, making them essential for saving operating systems, software, documents, photos, videos, and other important files.

Storage devices are generally classified into three main categories. They are

1. Internal Storage

Internal storage devices are built inside the computer and are used to store the operating system, software applications, and personal files.

- **Hard Disk Drive (HDD):**
  - Uses spinning magnetic disks (platters) to read/write data.
  - Typically provides high storage capacity at a lower cost.
  - Slower in speed compared to SSDs.
  - Common capacities range from 500 GB to several TB (terabytes).
- **Solid State Drive (SSD):**
  - Uses flash memory instead of spinning disks.
  - Much faster than HDDs in reading and writing data.
  - More durable (no moving parts) and energy-efficient.
  - Usually more expensive per GB than HDDs.
  - Common in modern laptops and desktops for faster boot and load times.

Advantages of Internal Storage:

- Faster access to data for system operations.
- Permanent installation inside the system.

- Lower risk of physical damage or theft compared to portable devices.

2. External Storage

External storage refers to portable devices that can be connected to the computer via USB ports or card readers. These devices are useful for data transfer, backup, and portability.

- **External Hard Drives:**
  - Same technology as internal HDDs or SSDs, but housed in a portable case.
  - Connects to the computer via USB, Thunderbolt, or eSATA.
  - Offers large capacity (up to several TB) and is ideal for backups and file transfers.
- **USB Flash Drives (Pen Drives):**
  - Small, portable devices that use flash memory.
  - Plug directly into USB ports.
  - Typically used for quick file transfers and portable storage.
  - Capacities range from **4 GB to 1 TB**.
- **SD Cards (Secure Digital Cards):**
  - Commonly used in cameras, smartphones, tablets, and some laptops.
  - Lightweight, small, and uses flash memory.
  - Requires an SD card slot or adapter to connect to computers.

Advantages of External Storage:

- Portable and easy to use.
- Convenient for sharing files between devices.
- Useful for data backups and expanding device storage.

3. Cloud Storage

Cloud storage refers to online services that store data on remote servers accessible via the internet. Data stored in the cloud can be accessed anytime, anywhere, using any internet-connected device.

- **Examples of Cloud Storage Services:**
  - Google Drive
  - Dropbox
  - Microsoft OneDrive
  - iCloud
- Cloud storage providers offer various plans:
  - Free limited storage (e.g., 5–15 GB).
  - Paid plans for higher storage capacity and advanced features.

Advantages of Cloud Storage:

- Accessibility from any device with internet access.
- Data backup and disaster recovery.
- Collaboration features for sharing and editing files in real-time.
- Reduces dependence on physical storage devices.

Limitations:

- Requires a stable internet connection.
- Security concerns if proper encryption and privacy settings are not applied.
- Might involve recurring subscription costs for large storage needs.

2.2.6 Software

Software refers to a set of coded instructions or programs that tell the hardware of a computer what actions to perform. While hardware is the physical part of a computer, software is the intangible component that controls and coordinates how the hardware operates.

Software is broadly classified into two main categories:

- **System Software**
- **Application Software**

A. System Software

System software is designed to manage the hardware resources of the computer and provide a platform for other software to run. It acts as a bridge between user applications and the computer's physical components.

Types of System Software

➢ **Operating System (OS)**

- The operating system is the core software that manages both hardware and software resources.
- It controls the CPU, memory, storage, input/output devices, and other hardware components.
- It provides a user interface (UI), such as a desktop or command line, to interact with the system.

Examples:

- Microsoft Windows
- macOS
- Linux
- Android (for mobile devices)

Key Functions:

- **Process Management** – Manages active applications and background processes.
- **Memory Management** – Allocates and tracks memory for each running program.
- **File System Management** – Manages data storage and retrieval.
- **Device Management** – Controls the use of hardware devices via drivers.
- **Security and Access Control** – Protects data and user privacy.

➢ **Device Drivers**

- Device drivers are specialized programs that allow the operating system to communicate with hardware devices.
- Every hardware component, like printers, graphic cards, keyboards, and USB drives, needs a driver to function properly.

Functions:

- Translate OS commands into device-specific signals.
- Ensure compatibility between the system and peripheral devices.
- Update regularly to support new features or fix bugs.

➢ **Utility Software**

- Utility software includes programs that help to maintain, analyze, optimize, and protect the computer system.
- These tools support smooth operation and enhance performance and security.

Examples of Utility Software:

- **Antivirus Programs** – Detect and remove malware.
- **Disk Cleanup Tools** – Free up space by removing unnecessary files.
- **Backup Software** – Helps create data copies for recovery.
- **Disk Defragmenters** – Organize fragmented data for faster access.
- **System Monitors** – Track system performance and resource usage.

B. Application Software

Application software is designed to help users perform specific tasks or activities. Unlike system software, it directly serves the end user and runs on top of the operating system.

Types of Application Software:

➢ **Word Processors**

- Used to create, edit, format, and print text-based documents.
- Common features include text formatting, spell check, grammar correction, and inserting images or tables.

Examples:

Microsoft Word ,Google Docs, LibreOffice Writer

Uses:

- Writing letters, reports, essays, and documentation.

➢ **Spreadsheets**

- Used for organizing data in rows and columns.
- Perform mathematical calculations, create graphs, and analyze data using formulas and functions.

Examples:

Microsoft Excel, Google Sheets, LibreOffice Calc

Uses:

- Budgeting, data analysis, statistical computation, business planning.

➢ **Web Browsers**

- Software used to access and view websites on the Internet.
- Interprets web content written in HTML, CSS, and JavaScript.

Examples:

Google Chrome, Mozilla Firefox,Microsoft Edge, Safari

Uses:

- Browsing information, watching videos, sending emails, online shopping, accessing social media.

Other Examples of Application Software

- **Presentation Software** – e.g., Microsoft PowerPoint for creating slideshows.
- **Database Management Systems (DBMS)** – e.g., MySQL, MS Access for managing structured data.
- **Graphics and Design Software** – e.g., Adobe Photoshop, Canva for creating images and layouts.
- **Media Players** – e.g., VLC Player for playing audio and video files.
- **Email Clients** – e.g., Microsoft Outlook, Thunderbird for managing email communication.

2.2.7 Communication Devices

Communication devices enable the computer to connect with other systems or networks.

1. Network Interface Card (NIC)

- A Network Interface Card (NIC) is a hardware component (either built-in or external) that allows a computer to connect to a network, typically a Local Area Network (LAN).
- It provides the physical interface between the computer and the network cable (Ethernet).
- NICs can be found in desktops, laptops, and servers.

Functions:
- Converts data from the computer into a format suitable for transmission over a network.
- Handles data packets (sending and receiving) through the network cable.
- Supports full-duplex communication, enabling simultaneous sending and receiving of data.

Types:
- Wired NIC – Uses an Ethernet cable (RJ-45 connector).
- Wireless NIC (Wi-Fi Card) – Allows wireless connections to Wi-Fi networks.

2. Modem (Modulator-Demodulator)
- A Modem is a device that allows computers to connect to the internet over telephone lines or cable systems.
- It works by converting digital signals from the computer into analog signals for transmission (modulation) and vice versa (demodulation) for reception.

Functions:
- Establishes connection to an Internet Service Provider (ISP).
- Sends and receives data through analog telephone or digital cable lines.
- Provides access to the World Wide Web (WWW) and other online services.

Types:
- **DSL Modem** – Connects via telephone lines.
- **Cable Modem** – Connects via TV cable lines.
- **Fiber Modem (ONT)** – Used for fiber-optic internet connections.

3. Wi-Fi Adapter
- A Wi-Fi Adapter is a device that allows a computer or laptop to connect wirelessly to a Wi-Fi network.
- It may be built into modern computers or added via USB dongles or internal cards.

Functions:
- Detects and connects to nearby wireless routers.
- Enables wireless internet access without the need for cables.
- Supports encryption protocols like WPA2 or WPA3 for secure communication.

Use Cases:
- Home internet browsing
- Wireless file sharing
- Accessing cloud storage and online services

4. Bluetooth Module
- A Bluetooth Module enables short-range wireless communication between the computer and other Bluetooth-enabled devices, such as smartphones, printers, speakers, and headsets.
- Operates over a short range (typically 10 meters) using radio frequency.

Functions:
- Transfers files between devices without the need for internet or cables.
- Connects peripheral devices like wireless keyboards, mice, and headphones.
- Uses low power, making it suitable for mobile and portable devices.

Bluetooth Versions:
- Newer versions like Bluetooth 5.0 offer higher speed and greater range compared to earlier versions.

Importance of Communication Devices

- Enable networking and internet access for browsing, emailing, cloud computing, and remote work.
- Support real-time communication through voice and video calls (e.g., Zoom, Skype).
- Facilitate file sharing and resource sharing such as printers and storage across networks.
- Make wireless and mobile computing more flexible and user-friendly.
- Essential in Internet of Things (IoT) devices and smart technologies.

2.3 COMPUTER ARCHITECTURE:

Computer Architecture refers to the design and organization of a computer's essential components, and how they work together to execute instructions. It defines the structure, behavior, and interaction of the hardware components of a computer system. Computer architecture focuses on how the CPU, memory, input/output devices, and buses are arranged and communicate with each other to perform tasks efficiently.It serves as a blueprint for building computer systems, much like architecture is a plan for constructing buildings.

2.3.1 Components of Computer Architecture

A typical computer architecture is based on the Von Neumann architecture, which includes the following main components:

1. Input Unit
2. Output Unit
3. Memory Unit
4. Central Processing Unit (CPU)
   - Arithmetic Logic Unit (ALU)
   - Control Unit (CU)
   - Registers
5. System Bus
   - Data Bus
   - Address Bus
   - Control Bus

2.3.2 Functional Units and Their Roles

1. Input Unit

The Input Unit is a vital component of the computer system responsible for receiving data and instructions from the user or the external environment and delivering them into the system for processing. It acts as the first step in the data processing cycle and serves as the gateway between the user and the computer.

Functions of the Input Unit

1. Receiving Data and Instructions

- The input unit accepts raw data, commands, and instructions entered by the user.
- These inputs can be in the form of text, numbers, audio, images, or signals.
- It serves as the entry point for any user interaction with the system.

2. Converting to Machine-Readable Form

- Computers can only understand binary language (0s and 1s).
- The input unit translates or encodes human-readable inputs into a digital format that the computer can understand and process.

3. Sending Data to Memory or CPU

- After conversion, the input unit transmits the data to the main memory (RAM) or directly to the CPU for immediate processing.

- This ensures the computer has the required data to perform operations.

Importance of the Input Unit

- It initiates communication between the user and the computer.
- Allows control over computing operations by providing necessary commands.
- Ensures accuracy by allowing verification before data is processed.
- Supports automation by accepting data from sensors or other machines in industrial systems.

2. Output Unit

The Output Unit is responsible for converting the processed data from the computer into a human-understandable format. It is the final stage of the data processing cycle, where the user receives the outcome of their input and instructions.

Functions of the Output Unit

1. **Converts Machine Output to Human-Readable Form**
   - The computer processes data in binary code (0s and 1s).
   - The output unit converts this digital information into text, graphics, audio, or visuals that humans can understand.
2. **Delivers Final Results to the User**
   - After processing is complete, the output unit displays or presents the final results to the user.
   - This could be on a screen, through sound, or as a printed document.

Importance of the Output Unit

- Completes the communication cycle between the computer and the user.
- Displays the results of data processing, making it possible to understand and act on information.
- Enhances user interaction through visual and audio feedback.

3. Memory Unit

The Memory Unit is a fundamental part of the computer system that is responsible for storing data and instructions. It plays a critical role in ensuring the smooth operation of the system by temporarily holding active data and permanently saving essential instructions and files.

The Memory Unit performs the following core tasks:

- **Stores Data and Instructions Temporarily or Permanently -** It holds data, instructions, and results either temporarily (volatile) or permanently (non-volatile) based on the type of memory.
- **Supports System Operations -** It is essential for running programs, managing files, and supporting the operating system in handling tasks efficiently.

Memory is mainly divided into two categories:

- **Primary Memory**

  Also known as main memory, this type of memory is directly accessible by the Central Processing Unit (CPU). It is faster than secondary storage but generally limited in capacity.

a. RAM (Random Access Memory)

- Volatile memory – loses its data when the computer is turned off.
- Stores data and instructions currently in use by the CPU.
- Acts as the working memory of the computer.
- The more RAM a system has, the better it can handle multitasking.

b. ROM (Read Only Memory)
- Non-volatile memory – retains its data even when the system is powered off.
- Contains permanent instructions, such as the startup firmware or BIOS.
- Usually pre-programmed by the manufacturer and not easily altered.

Characteristics of Primary Memory:
- High speed (fast access time)
- Limited capacity (usually measured in GB)
- Temporary and permanent storage roles (RAM = temporary, ROM = permanent)
- Essential for running applications and booting the computer

➢ **Secondary Memory**

Secondary storage provides long-term data storage. It is non-volatile, meaning data remains saved even when the system is powered down. It is not directly accessible by the CPU and must pass through I/O systems.

Examples:
- **Hard Disk Drive (HDD):**
  - Uses rotating magnetic disks to read/write data.
  - Offers high capacity at a lower cost.
  - Slower than RAM or SSDs.
- **Solid State Drive (SSD):**
  - Uses flash memory with no moving parts.
  - Faster and more reliable than HDDs.
  - Common in modern laptops and desktops for fast system boot and loadingtimes.
- **Optical Discs (CD/DVD):**
  - Use laser technology to store and retrieve data.
  - Mostly used for media storage, backups, or software distribution.
- **USB Flash Drives and SD Cards:**
  - Portable and widely used for transferring files.
  - Use flash memory, and are handy for backups and portability.

Characteristics of Secondary Storage:
- Large capacity (can be in TBs)
- Slower than primary memory
- Suitable for data archiving, storage of documents, media, and software
- Non-volatile – retains data without power

Functions of the Memory Unit

The memory unit performs several key functions in a computer system:
- **Stores Data Currently Being Processed** - Active data required for immediate operations is held in RAM for quick access by the CPU.
- **Holds Programs and the Operating System** - The OS, drivers, and active applications are stored in memory to be accessed and executed efficiently.
- **Retains Results Before Sending Them to the Output Unit -** Temporary storage of processed results before they are sent to output devices like monitors or printers.

The Memory Unit is essential for both the performance and functionality of a computer system. It manages the flow of data between the CPU, storage, and I/O devices. While Primary Memory (RAM and ROM) ensures fast access for active operations, Secondary Storage provides long-term data preservation. Together, these memory systems support all computing processes—from basic boot-up to complex software execution.

4. Central Processing Unit (CPU)

The CPU is the brain of the computer. It processes instructions and manages the flow of data.

a. Arithmetic Logic Unit (ALU)
- Performs all arithmetic operations (addition, subtraction, etc.).
- Handles logical operations (comparisons, AND, OR, NOT).

b. Control Unit (CU)
- Directs the operations of all other units.
- Fetches, decodes, and executes instructions in order.

c. Registers
- Small high-speed storage areas within the CPU.
- Temporarily hold data, instructions, and memory addresses.

2.4 SYSTEM BUS:

The System Bus is a critical communication pathway in a computer system that allows different components such as the CPU (Central Processing Unit), main memory (RAM), and input/output (I/O) devices to exchange information. It acts like a highway through which data travels within the computer. The efficiency and speed of the system bus directly affect the overall performance of the computer.

The system bus is generally made up of three types of buses: Data Bus, Address Bus, and Control Bus. Each plays a specific role in handling and directing communication between hardware components.

➢ **Data Bus**
- **Function**: The data bus is responsible for carrying actual data between the CPU, memory, and other peripherals.
- **Bidirectional**: The data bus can carry data to and from the CPU.
- **Width (Size)**: The number of bits the data bus can carry at one time is known as its **width** (e.g., 8-bit, 16-bit, 32-bit, 64-bit). A wider data bus means more data can be transferred at once, leading to better performance.
- **Example**: If the data bus is 32-bit wide, it can transfer 32 bits of data simultaneously.

➢ **Address Bus**
- **Function**: The address bus carries memory addresses from the CPU to other components, mainly memory. These addresses specify where data should be read from or written to.
- **Unidirectional**: Unlike the data bus, the address bus is usually **one-way**—from CPU to memory or I/O.
- **Width and Addressing**: The width of the address bus determines the maximum amount of memory the CPU can address. For example:
  - A 16-bit address bus can address 2^{16} = 65,536 memory locations.
  - A 32-bit address bus can address 2^{32} = 4,294,967,296 locations (4 GB of memory).

➢ **Control Bus**
- **Function**: The control bus carries control signals from the CPU to coordinate all operations of the computer. These signals manage data flow and timing among components.
- **Signals include**:
  - **Read** signal: Tells memory to send data to the CPU.
  - **Write** signal: Tells memory to receive data from the CPU.
  - **Clock signals**: Help synchronize operations.
  - **Interrupt requests**: Allow devices to signal the CPU that they need attention.

- **Bidirectional or Unidirectional**: Depending on the specific signal, the control bus can be one-way or two-way.

Importance of the System Bus

- The system bus is essential for CPU and memory communication.
- It helps the CPU to send commands, fetch data, and execute instructions by allowing access to memory and I/O devices.
- Performance Bottleneck: If the bus speed is slow, it becomes a bottleneck for the CPU, limiting system performance despite a fast processor.

2.5 SUMMARY:

In this lesson, we examined the core components that make up a computer system and how they work together to perform computing operations. Input and output devices facilitate interaction between the user and the system, while the Central Processing Unit (CPU) executes instructions through its Arithmetic Logic Unit (ALU), Control Unit (CU), and registers. Memory was explored in its two forms: primary (RAM and ROM) for fast, temporary access, and secondary storage (HDDs, SSDs, cloud) for long-term data retention. We also learned about software, distinguishing between system software that manages the system and application software that enables specific user tasks. Communication devices such as NICs, modems, and Wi-Fi adapters were discussed for their role in networking and internet connectivity.

2.6 KEY TERMS:

1. Input Devices, Central Processing Unit (CPU),Primary Memory (RAM/ROM),System Software, Communication Devices

2.7 REVIEW QUESTIONS:

1. What are the main components of a computer system? Briefly explain their functions.
2. How does the Central Processing Unit (CPU) process data using ALU and CU?
3. Differentiate between primary memory and secondary storage with suitable examples.
4. What is the role of system software in a computer system?
5. Explain the function and importance of communication devices in networking.

2.8 SUGGESTED READINGS:

1. “Computer Fundamentals” by P.K. Sinha – BPB Publications
2. “Fundamentals of Computers” by V. Rajaraman – PHI Learning

Dr.G. Neelima

LESSON- 3

HARDWARE, SOFTWARE, NETWORK TOPOLOGIES, AND STORAGE DEVICES

OBJECTIVES:

The objectives of the lesson are

1. Understand the role and types of computer hardware.
2. Differentiate between system, application, and programming software
3. Explain various network topologies
4. Compare storage devices and their uses

STRUCTURE:

3.1 Introduction

3.2 Internal vs External Hardware
- **3.2.1 Internal Hardware**
- **3.2.2 External Hardware**
- **3.2.3 Hardware Functionality Layers**
- **3.2.4 Embedded Hardware**
- **3.2.5 Upgradability and Compatibility**
- **3.2.6 Hardware Performance Factors**
- **3.2.7 Hardware Maintenance and Care**
- **3.2.8 Hardware in Modern Technology**

3.3 Software and Types of Software
- **3.3.1 System Software**
- **3.3.2 Application Software**
- **3.3.3 Programming Software**

3.4 Network Topology
- **3.4.1 Bus Topology**
- **3.4.2 Star Topology**
- **3.4.3 Ring Topology**
- **3.4.4 Mesh Topology**
- **3.4.5 Tree Topology**
- **3.4.6 Hybrid Topology**

3.5 Storage Devices
- **3.5.1 Hard Disk Drive (HDD)**
- **3.5.2 Solid State Drive (SSD)**
- **3.5.3 USB Flash Drives**
- **3.5.4 Secure Digital (SD) Cards**
- **3.5.5 Optical Discs**
- **3.5.6 External Hard Drives**
- **3.5.7 Cloud Storage**
- **3.5.8 Key Considerations When Choosing a Storage Device**

3.6. Summary

3.7 Key Terms

3.8 Self Assessment Questions

3.9 Suggestive Readings

3.1 INTRODUCTION:

Computers are an important part of our daily lives, helping us in areas like education, work, healthcare, and entertainment. To understand how computers function well and perform tasks quickly, it's important to know about two key parts: hardware and software. Hardware includes the physical parts of a computer, like the CPU, motherboard, and storage devices.

Software is a set of instructions that tells the hardware what to do. This chapter explains different hardware parts, including those inside and outside the computer, and how they work together. It also covers types of software such as system software, application software, and programming tools, and how they help in using the computer. The chapter also introduces network topologies, which show how computers connect and share information. Lastly, it discusses storage devices, including hard drives, SSDs, USB drives, and cloud storage. Learning about these topics helps us understand how computers work and how to use them better.

3.2 INTERNAL VS EXTERNAL HARDWARE:

Internal hardware refers to the components inside the computer case (like the motherboard, CPU, RAM, and power supply) that are essential for the system to function and process data. External hardware includes devices connected to the outside of the computer (like the monitor, keyboard, mouse, and USB drives) that help users interact with the system or expand its capabilities.

3.2.1 Internal Hardware

These components are located inside the computer case and are crucial for core operations.

- **Motherboard**: The main circuit board that holds the CPU, memory, and connectors for other hardware.
- **Graphics Processing Unit (GPU)**: Specialized processor for rendering images and videos.
- **Power Supply Unit (PSU)**: Converts electricity from a wall outlet into usable power for components.
- **Cooling Systems**: Includes heat sinks and fans to maintain safe temperatures.

3.2.2 External Hardware

These are outside the system unit and help in user interaction or connectivity.

- **Monitors, Keyboards, Mouse**: Devices used for user input/output.
- **External Drives**: Devices like external SSDs or HDDs used for extra storage.
- **Peripheral Ports**: USB, HDMI, and audio ports that allow hardware connections.

3.2.3 Hardware Functionality Layers

Hardware is structured in layers that work together:

- **Electrical Layer**: Circuits, voltage control, and signals used for communication.
- **Mechanical Layer**: Physical structures and bodies of components.
- **Logical Layer**: Microchips and processors that perform calculations and processing.

Each layer is essential for ensuring smooth interaction between hardware and software.

3.2.4 Embedded Hardware

Some hardware is integrated into other devices or systems. This is called embedded hardware, and it is designed to perform specific tasks.
Examples include:

- Microcontrollers in washing machines
- Embedded processors in smart TVs
- Sensors in fitness trackers

These systems are usually programmed to run fixed applications with minimal user control.

3.2.5 Upgradability and Compatibility

Unlike software, hardware has physical limitations. When updating or upgrading hardware, two important factors are:

- **Compatibility**: New components must be supported by existing systems (e.g., motherboard compatibility with new RAM or CPU).
- **Form Factor**: Refers to the physical size and layout (e.g., ATX for motherboards).

3.2.6 Hardware Performance Factors

Several factors influence hardware performance:

- **Clock Speed** (GHz): Measures the speed at which a processor performs operations.
- **Cache Memory**: Small memory within the CPU that speeds up access to frequently used data.
- **Bandwidth**: The amount of data a device can process in a given time.
- **Latency**: The delay before data transfer begins following an instruction.

3.2.7 Hardware Maintenance and Care

Proper maintenance ensures hardware remains functional and lasts longer:

- Regular dust cleaning to prevent overheating
- Safe shutdown procedures to protect storage devices
- Updating drivers to ensure optimal performance
- Monitoring temperature and health using diagnostic tools

3.2.8 Hardware in Modern Technology

Today's hardware extends beyond desktops and laptops. It powers devices like Smartphones, Drones, Self-driving cars, Smart home gadgets, Wearables and healthcare monitors. As technology evolves, hardware becomes more compact, efficient, and powerful.Hardware is the physical heart of any computing system. From simple circuit boards to advanced processing units, it supports all digital operations. Understanding its structure, role, and care is key to efficient and long-lasting computing.

3.3 SOFTWARE AND TYPES OF SOFTWARE:

Software is a collection of coded instructions, programs, and data that direct the hardware on what tasks to perform. While hardware is the body of a computer, software acts as the mind that guides its actions. Software enables users to interact with a computer system and use it for various operations, ranging from basic typing to complex data processing.Software is the driving force behind all computer operations. It tells hardware what to do and enables users to perform specific tasks, from typing a document to analyzing data or communicating across the globe. As the digital world expands, software continues to evolve becoming smarter, more powerful, and more accessible.Unlike hardware, software is intangible, meaning it cannot be physically touched but is essential for system operation.

Software is generally classified into three major categories based on its function. They are

3.3.1 System Software

System software is a category of software that acts as a bridge between the hardware and the user. It manages and controls hardware operations, provides the environment for application software to run, and ensures that the entire system functions smoothly. Without system software, the hardware would not know how to operate or interact with other components.

Types of System Software

System software is mainly divided into the following categories:

1. Operating System (OS)

The Operating System is the core system software that manages both hardware and software resources of a computer. It acts as the main interface between the user and the hardware. All other programs need the operating system to function.

Functions of an Operating System:

- **Resource Management:** Allocates CPU, memory, and input/output devices efficiently.
- **Process Management:** Manages running applications and background tasks.
- **File System Management:** Organizes and controls file storage and retrieval.
- **User Interface:** Provides graphical or command-line interfaces to interact with the system.
- **Security and Access Control:** Protects data and system files with passwords, permissions, and firewalls.
- **Multitasking:** Allows multiple programs to run at the same time without conflict.

Popular Examples:

- **Windows:** Widely used OS for PCs and laptops, developed by Microsoft.
- **Mac OS:** Apple's operating system for Mac computers.
- **Linux:** Open-source OS often used in servers and tech environments.
- **Android:** Google's OS for mobile phones and tablets.

2. Device Drivers

Device drivers are small programs that allow the operating system to communicate with hardware devices.

Without drivers, hardware components like printers, keyboards, and graphic cards would not work correctly. Every device connected to a computer has a specific driver.

Functions of Device Drivers:

- Translates OS commands into device-specific instructions.
- Enables compatibility between hardware and software.
- Allows hardware customization and advanced features (e.g., printer settings).
- Improves performance by ensuring efficient communication with the operating system.

Examples of Hardware Requiring Drivers:

Printers**,** Scanners**,** Graphic cards**,** Sound cards **,**USB devices**,** Network adapters

Most modern operating systems come with a large collection of built-in drivers and can automatically download and install them when new hardware is connected.

3. Utility Programs

Utility software includes tools that help maintain, manage, analyze, and protect the computer system. These programs run in the background and support the overall performance, security, and smooth functioning of the computer.

Common Types of Utility Programs:

- **Antivirus Software:** Scans and removes viruses, malware, and spyware.
  Example: Norton, Avast, Windows Defender
- **Disk Cleanup Tools:** Delete temporary files, cache, and junk files to free up space.
- **Backup Software:** Creates backup copies of data to prevent loss in case of failure.
- **Disk Defragmenter:** Reorganizes fragmented data on hard drives for faster access.
- **System Monitors:** Track CPU usage, memory usage, temperature, and other system parameters.
- **Compression Tools:** Reduce file sizes for easier storage or transfer.
  Example: WinRAR, 7-Zip

Importance of Utility Software:

- Improves system speed and efficiency.
- Prevents data loss.
- Keeps the computer secure and clean.
- Helps in troubleshooting and diagnosing issues.

System software is essential for running any computer system. It controls how the system starts, interacts with hardware, runs applications, and protects the device. Operating systems handle overall control, device drivers ensure communication with hardware, and utility programs maintain and optimize the system. Understanding these components helps in effectively managing and using a computer.

3.3.2 Application Software

Application software refers to programs that are designed to help users perform specific tasks or activities. Unlike system software, which operates the computer, application software allows users to interact with the system to carry out personal, educational, or business-related tasks.

Application software runs on top of the operating system and is tailored for end-user productivity, communication, design, data management, and more.

Key Features of Application Software

- Designed for specific user needs.
- User-friendly and often has graphical interfaces.
- Can be free, paid, or subscription-based.
- Often customizable based on tasks and preferences.

Types of Application Software and Their Uses

1. Word Processors

Purpose: Used to create, edit, format, and print text documents.

Common Features:

- Text formatting (bold, italic, underline)
- Spell check and grammar correction
- Page setup and printing options
- Inserting images, tables, and charts
- Saving documents in various formats (DOCX, PDF)

Examples:

Microsoft Word, Google Docs, LibreOffice Writer

Uses:

- Writing letters, essays, resumes
- Creating official documents and reports
- Drafting books, brochures, and newsletters

2. Spreadsheets

Purpose: Used for organizing data in rows and columns, performing calculations, and creating charts.

Common Features:

- Built-in formulas and functions (SUM, AVERAGE, IF, etc.)
- Data sorting and filtering
- Graphs and chart generation
- Pivot tables for data analysis

Examples:

Microsoft Excel, Google Sheets, LibreOffice Calc

Uses:

- Budgeting and accounting
- Business and financial analysis
- Recording student marks or employee records
- Creating invoices and performance reports

3. Web Browsers

Purpose: Used to access websites and online content through the internet.

Common Features:

- Viewing text, images, videos, and animations
- Bookmarking and tabbed browsing
- Privacy settings and browsing history
- Extensions and plugins for added functionality

Examples:

Google Chrome, Mozilla Firefox, Microsoft Edge, Safari

Uses:

- Browsing information and news
- Sending and reading emails
- Watching videos or accessing social media
- Conducting online transactions

4. Presentation Tools

Purpose: Used to create slideshows for presenting information visually during meetings, lectures, or events.

Common Features:

- Slide layouts with text, images, videos, and charts
- Transitions and animations
- Templates and design themes
- Presenter mode and slide notes

Examples:

Microsoft PowerPoint, Google Slides, Canva (for visually rich presentations)

Uses:

- Classroom lessons and educational content
- Business proposals and project pitches
- Event planning or product showcasing

5. Multimedia Players

Purpose: Used to play audio and video files.

Common Features:

- Support for multiple formats (MP3, MP4, AVI, MKV, etc.)
- Subtitle support
- Audio equalizer settings
- Playlist creation and media libraries

Examples:
VLC Media Player, Windows Media Player, KMPlayer
Uses:

- Watching movies and videos
- Listening to music or audio lectures
- Viewing recordings of events or classes

6. Photo Editing Tools
Purpose: Used to edit and enhance images.
Common Features:

- Cropping, resizing, and rotating images
- Adjusting brightness, contrast, and color
- Adding filters, text, or graphic elements
- Layer-based editing and effects

Examples:
Adobe Photoshop, GIMP (free and open-source), Canva (for simple graphic design)
Uses:

- Designing posters, flyers, and banners
- Retouching photos for photography or media
- Creating graphics for social media and marketing

Application software is essential for daily user interaction with computers and smart devices. Whether for education, business, creativity, or communication, these programs make computers functional and productive for end users. Their user-friendly nature allows even non-technical individuals to perform complex tasks with ease.

3.3.3 Programming Software

Programming software provides the tools needed for writing, testing, and maintaining code. These tools are used by developers to build both system and application software.
Examples:

- **Text Editors:** Notepad++, Visual Studio Code
- **Compilers:** Convert high-level programming languages into machine code.
  Examples: GCC (for C/C++), Java Compiler
- **Debuggers:** Help identify and fix code errors.
- **Interpreters:** Run code line-by-line, commonly used in scripting languages like Python.

Software must be installed or run directly to perform tasks by using system resources like CPU and memory. It comes in various license types such as freeware, open source, shareware, and proprietary, each defining how the software can be used or shared. Software is essential for managing hardware, enabling user interaction, running applications, and automating tasks. It is developed using the Software Development Life Cycle (SDLC) and regularly updated for bug fixes, new features, and security. Emerging trends include cloud-based software, mobile apps, AI and machine learning tools, SaaS platforms, and VR/AR technologies.

3.4 NETWORK TOPOLOGY:

Network topology refers to the arrangement of various elements (links, nodes, etc.) in a computer network. It defines how devices are interconnected and how data is transmitted among them. Understanding network topologies is crucial for designing efficient and robust networks.
Different Network topologies are represented in the below figure

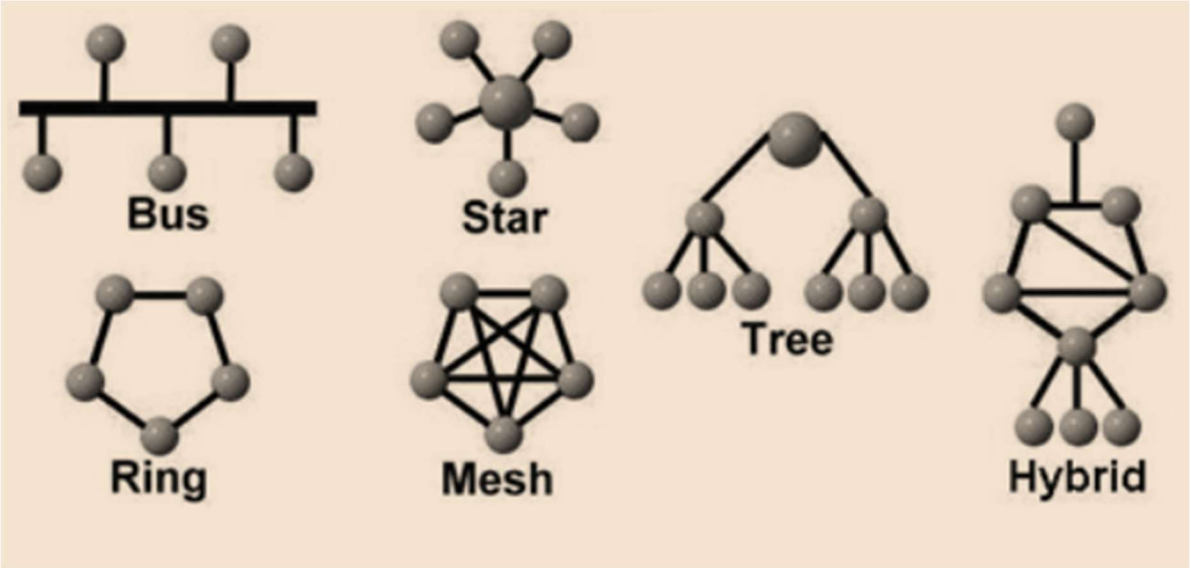


Figure 3.1 Network Topologies

The main types of network topologies include:

3.4.1 Bus Topology

In a bus topology, all devices (nodes) are connected to a single central cable known as the bus or backbone. This cable carries data to all devices, but only the one with the matching address accepts the data.

How it Works:

When a device sends data, it travels along the bus in both directions and is received by all connected devices. Only the intended recipient processes it; others ignore it.

Advantages:

- Easy and inexpensive to set up for small networks.
- Requires less cable compared to other topologies.
- Ideal for temporary or small networks.

Disadvantages:

- The entire network stops working if the main cable fails.
- Data collisions are common, especially in larger networks.
- Difficult to troubleshoot.

Use Cases:

Used in small office or home networks, especially in earlier networking designs.

3.4.2 Star Topology

In a star topology, all devices are connected to a central node (hub or switch). This central node acts as a repeater for data flow.

How it Works:
Each device has a dedicated point-to-point link to the hub. The hub receives data from a device and forwards it to the destination device.
Advantages:
- Easy to install, manage, and troubleshoot.
- A failure in one device or cable does not affect the others.
- Centralized management makes it efficient.

Disadvantages:
- Failure of the central hub brings down the entire network.
- Requires more cabling than bus topology.

Use Cases:
Most common in modern LANs (Local Area Networks), especially in schools, offices, and homes.

3.4.3 Ring Topology
In a ring topology, each device is connected to exactly two other devices, forming a closed loop (ring).
How it Works:
Data travels in a unidirectional or bidirectional circle until it reaches its destination. Each device has a repeater to forward data to the next.
Advantages:
- All devices have equal access to the network.
- Performs better than bus under heavy load.

Disadvantages:
- A failure in any single connection can disrupt the entire network.
- Adding or removing devices can be difficult.

Use Cases:
Used in wide-area networks (WANs) and some LANs with token ring technology.

3.4.4 Mesh Topology
In a mesh topology, each device is connected to every other device on the network.
How it Works:
Data can be transmitted along multiple paths. Mesh topology supports both full mesh (every node is connected to every other) and partial mesh (some nodes are connected to multiple nodes).
Advantages:
- Very reliable and fault-tolerant.
- No data traffic issues due to multiple paths.
- Good for networks requiring high availability.

Disadvantages:
- Expensive and complex to set up due to cabling and ports.
- Difficult to manage in large networks.

Use Cases:
Used in critical communication setups like military, industrial, and large-scale wireless networks.

3.4.5 Tree Topology
A tree topology is a combination of star and bus topologies. It consists of multiple star-configured networks connected to a single bus (backbone).

How it Works:
Devices are grouped into star networks and connected via a central bus. The hierarchy supports central control and segmentation.
Advantages:
- Scalable and easy to expand.
- Fault detection is easier and isolation is possible.
- Hierarchical structure allows for better management.

Disadvantages:
- Failure in the main backbone affects the entire network.
- Requires more cabling and configuration.

Use Cases:
Used in large organizations, school campuses, and enterprise networks.

3.4.6 Hybrid Topology
A hybrid topology combines two or more different types of topologies (e.g., star + mesh, bus + ring).
How it Works:
The combination is tailored to meet specific network needs, using the best features of each topology type.
Advantages:
- Highly flexible and scalable.
- Can be customized based on performance, cost, and reliability needs.
- Fault isolation is easier.

Disadvantages:
- Can be complex and costly to design and maintain.
- Requires skilled management and setup.

Use Cases:
Used in large enterprises, data centers, and telecommunications where diverse network demands exist.

3.5 STORAGE DEVICES:

Storage devices are integral components of computer systems, responsible for saving and retrieving digital data. They ensure that information, ranging from the operating system to personal files, remains accessible and intact even when the computer is powered off. Over the years, various storage devices have been developed, each tailored to specific needs and technological advancements.

3.5.1 Hard Disk Drive (HDD)
A Hard Disk Drive (HDD) is a permanent data storage device used in computers, laptops, servers, and external enclosures. It stores everything from the operating system, software programs, documents, and personal files like photos, videos, and music.
The HDD is a non-volatile storage device, which means it retains data even when the power is turned off. It has been widely used for decades and remains popular because of its large capacity and affordability.

Construction and Components
An HDD contains several mechanical and electronic parts inside a sealed metal case:
- **Platters**: These are circular magnetic disks made of glass or aluminum. They spin at high speeds (e.g., 5400 RPM or 7200 RPM).

- **Spindle**: Holds the platters and rotates them.
- **Read/Write Head**: A tiny magnetic sensor that floats just above the platter surface to read or write data.
- **Actuator Arm**: Moves the read/write head to the correct position over the platter.
- **Actuator**: Controls the movement of the actuator arm.
- **Interface Controller**: Connects the HDD to the computer (using SATA or IDE ports) and manages communication.

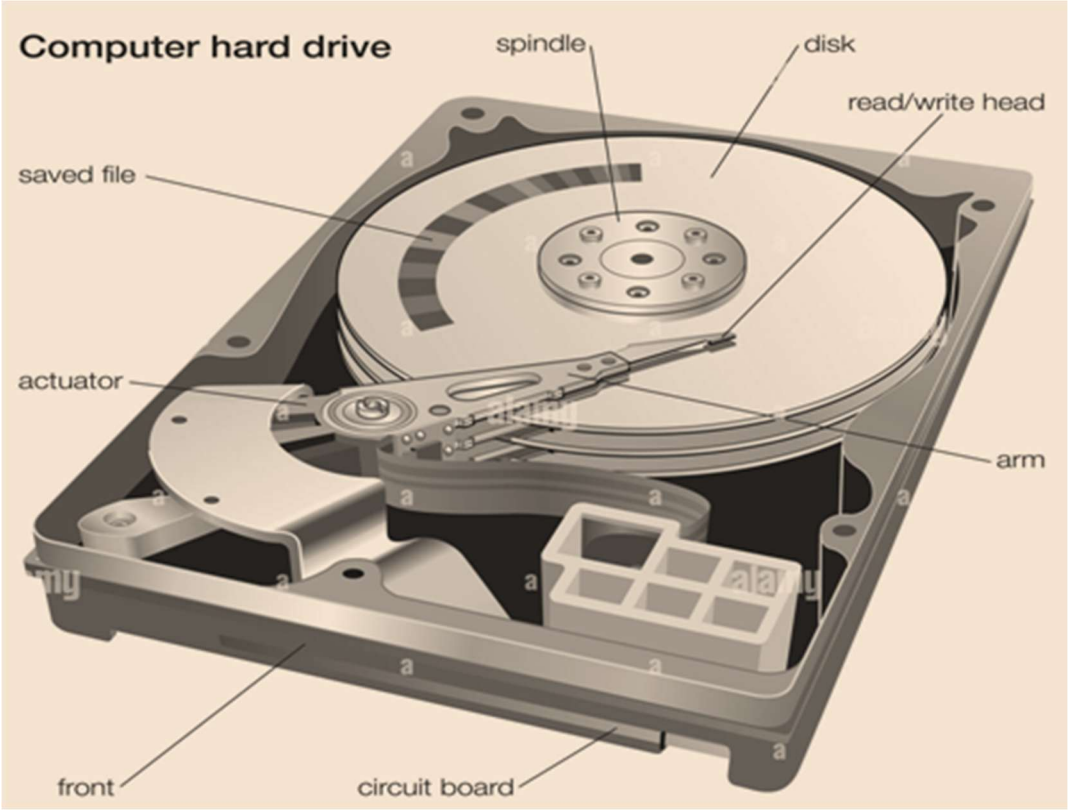


Figure 3.2 Components of HDD

How Does an HDD Work?

The HDD works like a record player, but with much greater precision and speed:

1. **Spinning the Platters**: When the computer is powered on, the platters begin spinning rapidly.
2. **Moving the Read/Write Head**: The actuator arm moves the head to the correct location above the spinning platter.
3. **Reading Data**: Data stored on the platter is in the form of magnetic patterns. As the head passes over these patterns, it detects changes in the magnetic field and converts them into digital data.
4. **Writing Data**: When saving files, the head changes the magnetic field on the platter's surface to represent new data. It writes 1s and 0s (binary code), which the computer understands.
5. **Data Transfer**: The interface controller sends and receives this data between the HDD and the computer's motherboard.

Functionality in Simple Terms

- Think of the HDD like a library with spinning shelves.
- The platters are the spinning shelves where books (data) are stored.

- The actuator arm and head are like a robotic librarian that moves to find the right book.
- When you open a file, the librarian reads the page (data) and gives it to you.
- When you save something, the librarian writes the page and puts it on the shelf.

Common Capacities and Performance

- **Storage Size**: Common sizes include 500 GB, 1 TB, 2 TB, and 4 TB or more.
- **Speed**: Measured in RPM (revolutions per minute). Most HDDs spin at 5400 RPM or 7200 RPM. Higher RPM = faster access.
- **Data Transfer Rate**: Around 80–160 MB/s depending on the model.

Advantages of HDD

- **Large Storage Capacity**: Ideal for storing big files and backups.
- **Cost-Effective**: Cheaper than SSDs per gigabyte.
- **Long Lifespan**: Can last many years if handled properly.

Disadvantages of HDD

- **Slower than SSDs**: Mechanical parts take time to move and access data.
- **Susceptible to Damage**: Can be damaged by drops, shocks, or wear over time.
- **Noisy and Generates Heat**: Due to moving parts.

Where HDDs are Used

- Desktop and laptop computers (especially budget models)
- External drives for backups
- Servers with large storage needs
- CCTV and surveillance systems for recording video

3.5.2 Solid State Drive (SSD)

A Solid State Drive (SSD) is a modern data storage device that uses flash memory to store information. Unlike traditional Hard Disk Drives (HDDs), SSDs have no moving parts, resulting in faster data access, reduced latency, and enhanced durability.

Components of an SSD

1. **Flash Memory Chips:** These non-volatile memory chips store data persistently without the need for power. They are organized into memory cells that hold bits of data.
2. **Controller:** This embedded processor manages read and write operations, error correction, and wear leveling, ensuring efficient and reliable performance.
3. **Cache (Optional):** Some SSDs include DRAM cache to temporarily store data, enhancing speed and responsiveness.

How SSDs Work

When data is written to an SSD, the controller directs it to appropriate flash memory cells, ensuring even distribution to prevent wear. For read operations, the controller quickly retrieves data from the cells and sends it to the computer. The absence of moving parts allows SSDs to access data almost instantaneously, leading to faster boot times and quicker file transfers.

Advantages of SSDs

- **Speed:** SSDs offer significantly faster read and write speeds compared to HDDs, enhancing overall system performance.
- **Durability:** With no mechanical components, SSDs are more resistant to physical shock and less prone to failure.
- **Energy Efficiency:** SSDs consume less power, leading to longer battery life in portable devices.

Diagram of SSD Architecture
For a visual representation of SSD components and their interconnections, refer to the following diagram:

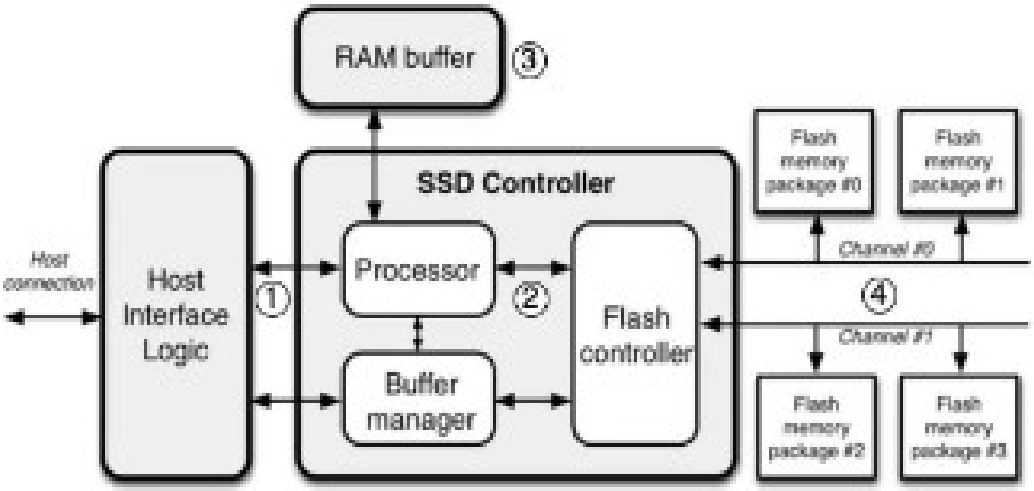


Figure 3.3 SSD Architecture

This diagram illustrates the relationship between the controller, flash memory, and other essential components within an SSD. Understanding the structure and functionality of SSDs highlights their advantages over traditional storage methods, making them a preferred choice in modern computing for enhanced performance and reliability.

3.5.3 USB Flash Drives

Also known as thumb drives or pen drives, these portable devices use flash memory to store data. They connect via USB ports and are widely used for transferring files between devices. Their small size and ease of use have made them a popular choice for personal and professional data transfer needs.

3.5.4 Secure Digital (SD) Cards

SD cards are compact memory cards commonly used in portable devices like digital cameras, smartphones, and tablets. They come in various capacities and speeds, catering to different performance requirements. MicroSD cards, a smaller variant, are prevalent in mobile devices due to their minimal size.

3.5.5 Optical Discs

This category includes CDs, DVDs, and Blu-ray discs. Data is written and read using laser technology. While once a primary medium for media distribution and backups, their usage has declined with the advent of more efficient storage solutions. However, they are still utilized for specific applications and archival purposes.

3.5.6 External Hard Drives

These are HDDs or SSDs encased in a portable enclosure, connecting to computers via interfaces like USB or Thunderbolt. They provide an excellent solution for backing up large volumes of data and expanding storage capacity without internal upgrades.

3.5.7 Cloud Storage

A modern solution where data is stored on remote servers accessed via the internet. Services like Dropbox, Google Drive, and OneDrive offer scalable storage options, enabling users to

access their data from any device with an internet connection. Cloud storage also facilitates collaboration and data synchronization across multiple devices.

3.5.8 Key Considerations When Choosing a Storage Device

- **Capacity Needs:** Assess the volume of data you intend to store to select a device with adequate space.
- **Speed Requirements:** For tasks requiring rapid data access, such as gaming or video editing, SSDs are preferable due to their superior speeds.
- **Portability:** If you need to transport data frequently, portable solutions like USB flash drives or external SSDs are ideal.
- **Durability:** SSDs and flash-based devices are more resilient to physical shocks compared to HDDs, making them suitable for mobile use.
- **Budget Constraints:** While SSDs offer enhanced performance, they come at a higher cost per gigabyte compared to HDDs.

3.6. SUMMARY:

This lesson explains the basic parts of a computer and how they work together. It covers internal hardware like the CPU and RAM, and external hardware like the keyboard and monitor. It also talks about embedded hardware found in smart devices and the importance of keeping hardware clean and working well. The lesson then explains software, which is the set of instructions that tells the hardware what to do. It describes system software, application software, and programming software with simple examples. Next, it introduces network topologies, which are ways computers connect and share data, such as bus, star, ring, mesh, tree, and hybrid topologies. Finally, it covers different storage devices like HDDs, SSDs, USB drives, and cloud storage, and explains what to consider when choosing one. Overall, the lesson helps students understand how computers work and how to use them better.

3.7 KEY TERMS:

Hardware, System Software, Application Software, Network Topology, Storage Devices

3.8 SELFASSESSMENT QUESTIONS:

1. What is the main difference between internal and external hardware components?
2. Explain how system software enables hardware to function.
3. Describe the structure and working of a Hard Disk Drive (HDD).
4. Identify and compare at least three different types of network topologies.
5. What factors should be considered when selecting a suitable storage device?

3.9 SUGGESTIVE READINGS:

1. "Computer Organization and Design" by David A. Patterson and John L. Hennessy – A comprehensive guide on computer hardware and software integration
.
2. "Operating System Concepts" by Abraham Silberschatz, Greg Gagne, and Peter B. Galvin – Essential reading on system software, particularly operating systems.

Dr. G. Neelima

LESSON- 4

MS WORD: DOCUMENT CREATION, FORMATTING, AND EDITING

OBJECTIVES:

The objectives of the lesson are

1. To understand how to create, save, and manage documents using Microsoft Word.
2. To learn how to format text, paragraphs, and pages for a professional look.
3. To explore editing tools like Find and Replace, Spell Check, and Track Changes.
4. To learn how to insert tables, images, and other elements to make documents more effective.

STRUCTURE:

4.1 Introduction to Microsoft Word

4.2 Document Creation
- 4.2.1 Creating a New Document
- 4.2.2 Working with Text
- 4.2.3 Saving a Document
- 4.2.4 Reviewing and Proofreading
- 4.2.5 Printing and Sharing Documents

4.3 Formatting
- 4.3.1 Types of Formatting in MS Word
- 4.3.2 Text (Character) Formatting
- 4.3.3 Paragraph Formatting
- 4.3.4 Page Formatting
- 4.3.5 Styles and Themes
- 4.3.6 Formatting Tools and Shortcuts
- 4.3.7 Best Practices in Formatting

4.4 Editing
- 4.4.1 Basic Text Editing Functions
- 4.4.2 Selecting Text
- 4.4.3 Find and Replace
- 4.4.4 Inserting and Deleting Text
- 4.4.5 Using the Clipboard
- 4.4.6 Spell Check and Grammar Tools
- 4.4.7 Track Changes and Comments
- 4.4.8 Editing with Navigation Pane
- 4.4.9 Advanced Editing Tools

4.5 Summary

4.6 Key Terms

4.7 Self Assessment Questions

4.8 Suggestive Readings

4.1 INTRODUCTION TO MICROSOFT WORD:

Microsoft Word is a powerful word processing application developed by Microsoft and is part of the Microsoft Office suite. It is designed to allow users to create, edit, format, and manage text-based documents easily and efficiently. Microsoft Word is widely used in academic, business, and personal settings due to its intuitive interface and vast range of features.Creating a document in MS Word is one of the most fundamental computer skills. Whether preparing a school assignment, writing a formal letter, drafting a report, or designing a brochure, MS Word provides all the tools necessary to produce professional and well-structured documents.

4.2 DOCUMENT CREATION:

Before starting document creation, the user must open the Microsoft Word application. This can be done through the following methods:

- **Start Menu**: Click the Start button, type "Word" into the search box, and click on Microsoft Word from the search results.
- **Desktop Shortcut**: Double-click on the Word icon if a shortcut is present on the desktop.
- **Taskbar or App Launcher**: Select Word if it is pinned to the taskbar or located in the list of installed applications.

Upon launching, the Start Screen will appear, offering options such as creating a Blank Document, using a Template, or opening an Existing File.

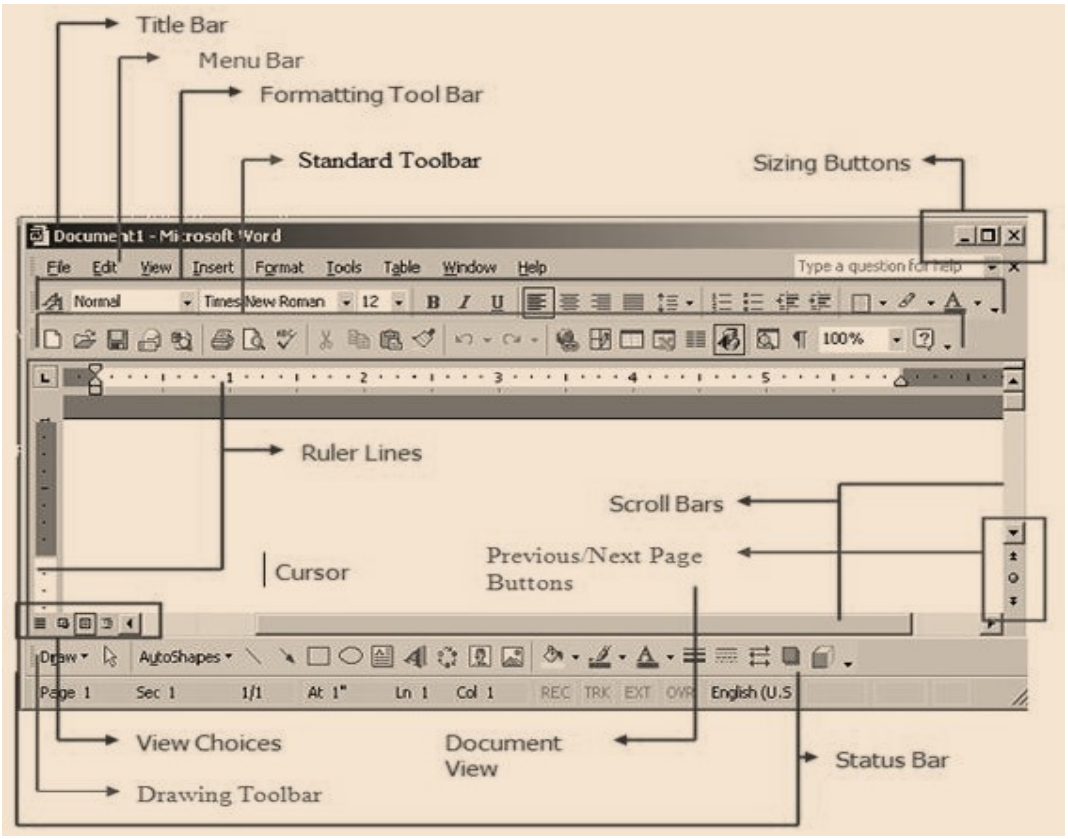


Figure 4.1 MS Word Toolbars

4.2.1 Creating a New Document

After launching the application, document creation can begin by choosing:

- **Blank Document**: This option opens a clean, white page, giving the user full control over layout, formatting, and design.
- **Templates**: These are pre-designed documents with built-in formatting and design elements, useful for resumes, letters, reports, flyers, and more. They help save time and maintain consistency.

Templates can be accessed from the built-in gallery or downloaded from Microsoft's online library.

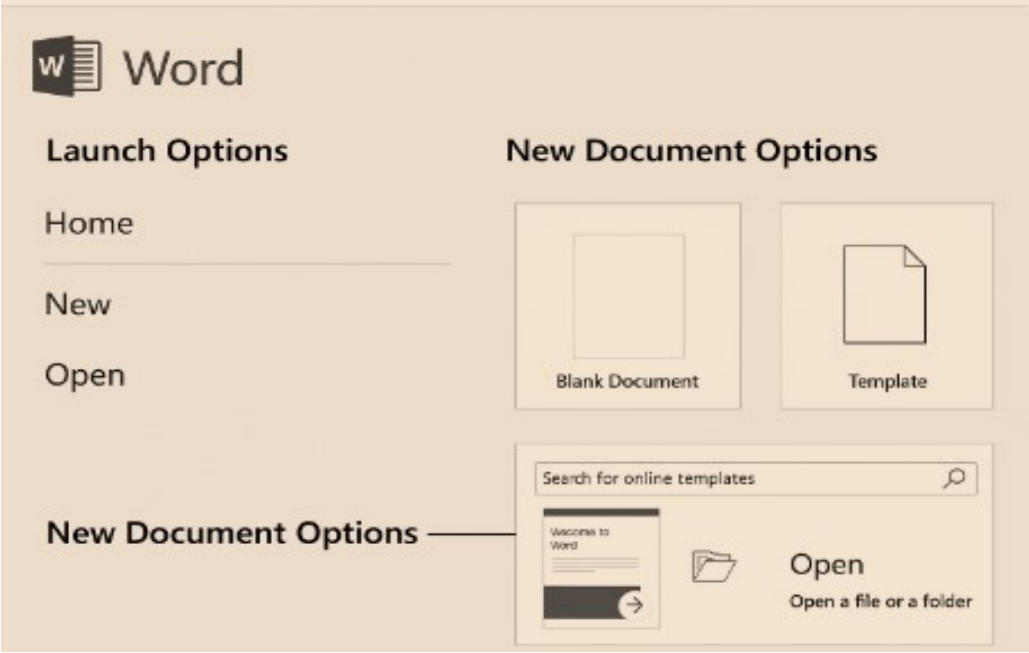


Figure 4.2 Creating a New Document

4.2.2 Working with Text

Once a new document is open, users can begin typing directly into the workspace. Microsoft Word supports:

- **Typing and Editing**: Simple typing, deleting, copying, and pasting of text.
- **Navigation**: Use the mouse or arrow keys to move through the document.
- **Selection**: Highlight text for formatting or editing using the mouse or keyboard shortcuts like Shift + Arrow keys.

Word also offers tools such as Undo, Redo, and Clipboard history for better text management.

Text Formatting

Text formatting is crucial for enhancing readability and visual appeal. MS Word provides a wide range of formatting tools in the Home tab, such as:

- **Font Style and Size**: Choose from various fonts like Times New Roman, Arial, Calibri, etc., and adjust the font size.
- **Font Effects**: Apply bold, italic, underline, strikethrough, subscript, and superscript.
- **Text Color and Highlighting**: Change font color or highlight specific text to emphasize.
- **Alignment**: Align text to the left, right, center, or justify to both sides.
- **Line and Paragraph Spacing**: Control spacing between lines and paragraphs for better structure.
- **Bullets and Numbering**: Create ordered (numbered) and unordered (bulleted) lists.

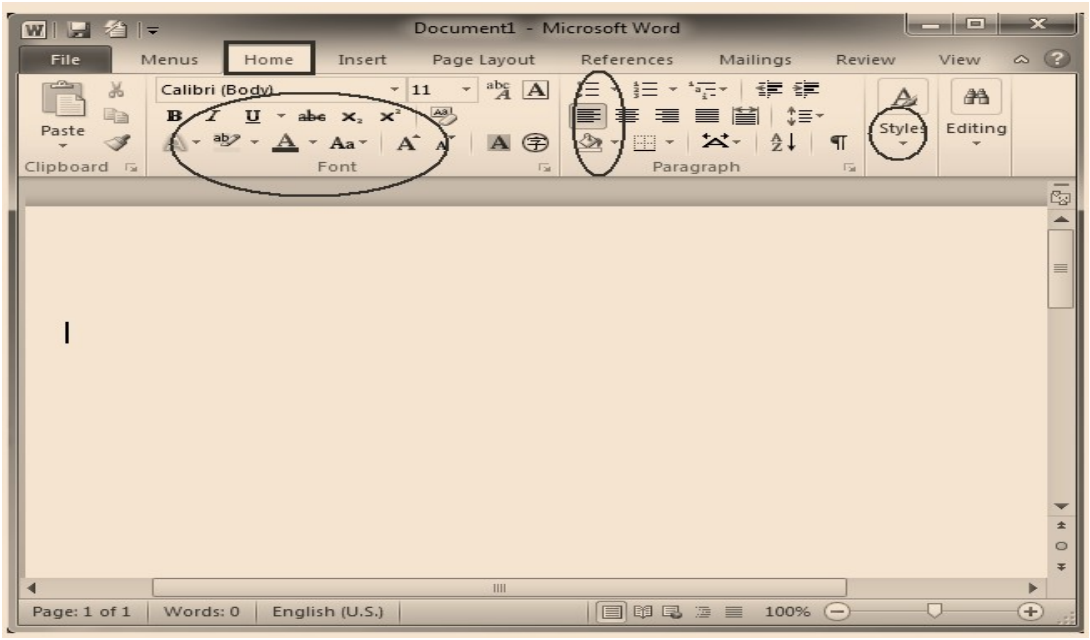


Figure 4.3 Text Formatting

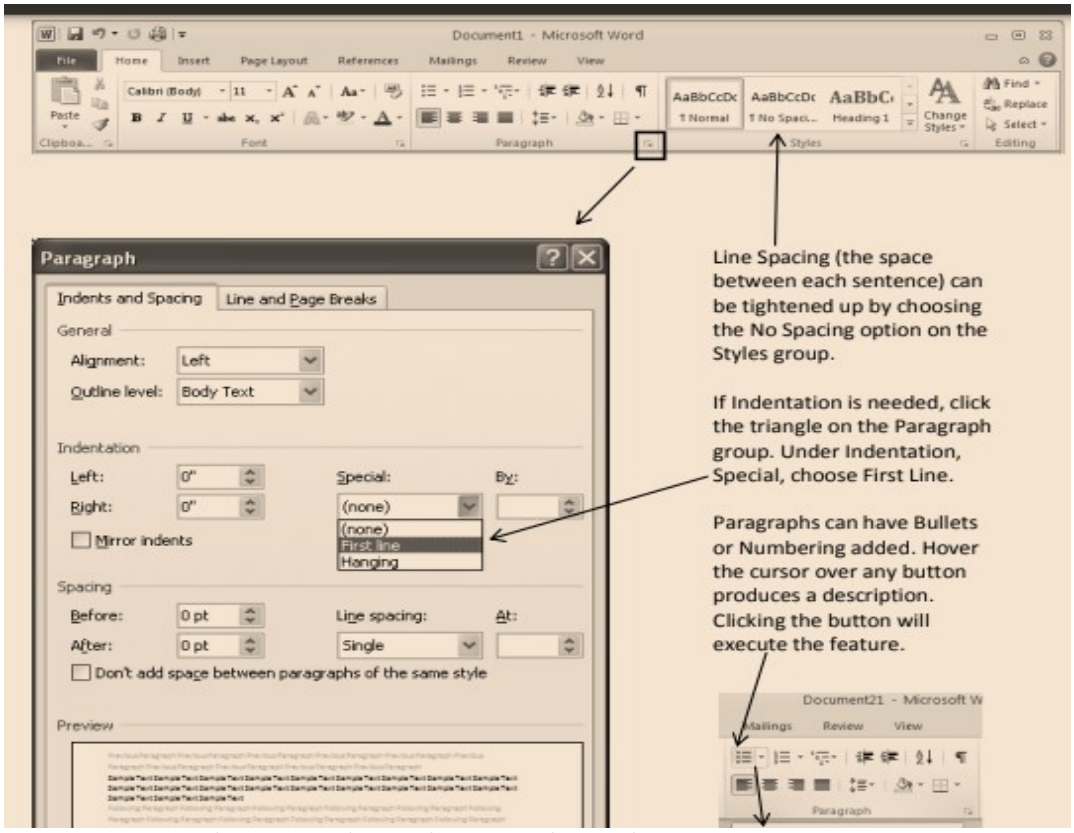


Figure 4.4 Line and Paragraph Spacing

These features allow users to structure content logically and improve overall presentation.

Paragraph and Page Layout

Microsoft Word also supports formatting at the paragraph and page level. Features include:

- **Indentation and Tabs**: Indent paragraphs to create emphasis or structure documents with multiple levels.
- **Styles**: Apply predefined heading and text styles for uniform formatting.
- **Page Orientation**: Choose between portrait or landscape layout.
- **Margins and Page Size**: Adjust page margins and choose standard page sizes such as A4 or Letter.
- **Columns and Breaks**: Organize text into multiple columns or insert section/page breaks.
-

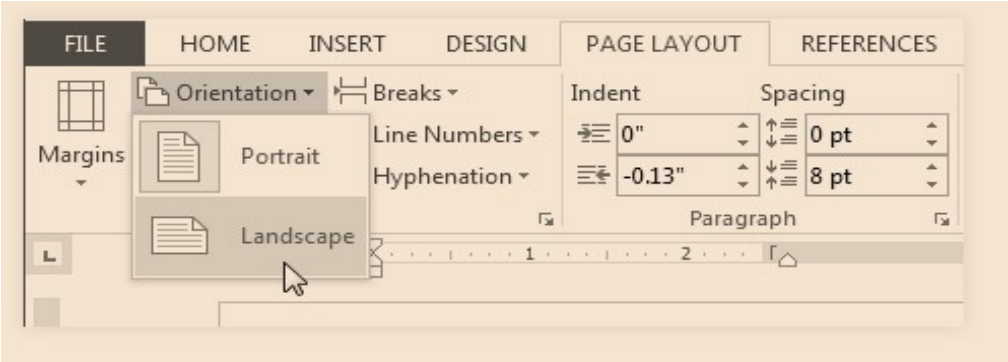


Figure 4.5 Page Orientation

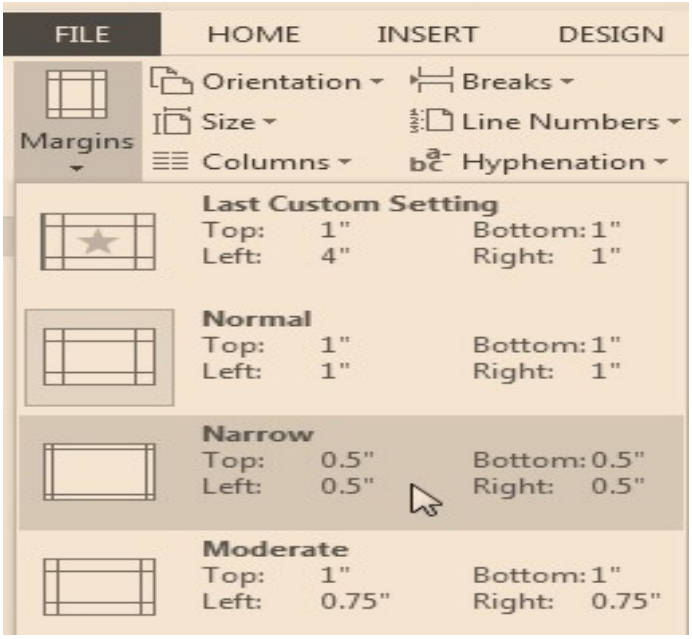


Figure 4.6 Margins and Page Size

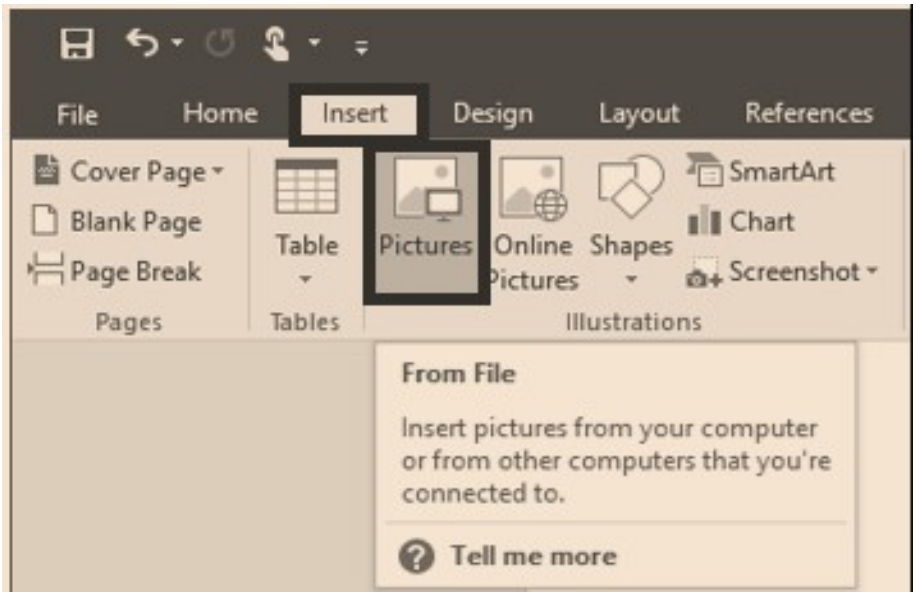
These tools help in producing well-organized and aesthetically pleasing documents.

Inserting Additional Elements

To enrich the document and support the text content, MS Word allows users to insert various elements through the Insert tab:

- **Tables**: Organize data into rows and columns.
- **Pictures and Online Images**: Add graphics from your computer or the web.

- **Shapes and SmartArt**: Insert diagrams and illustrations to convey information visually.
- **Charts**: Create bar graphs, pie charts, and other visual data representations.
- **Hyperlinks**: Link to websites, files, or other parts of the document.
- **Headers and Footers**: Include repeating text such as page titles, author names, or dates at the top or bottom of pages.
- **Page Numbers**: Automatically number pages for long documents.
- **Text Boxes and WordArt**: Insert styled text for titles and highlights.
- **Symbols and Equations**: Use mathematical and special symbols where needed.
-



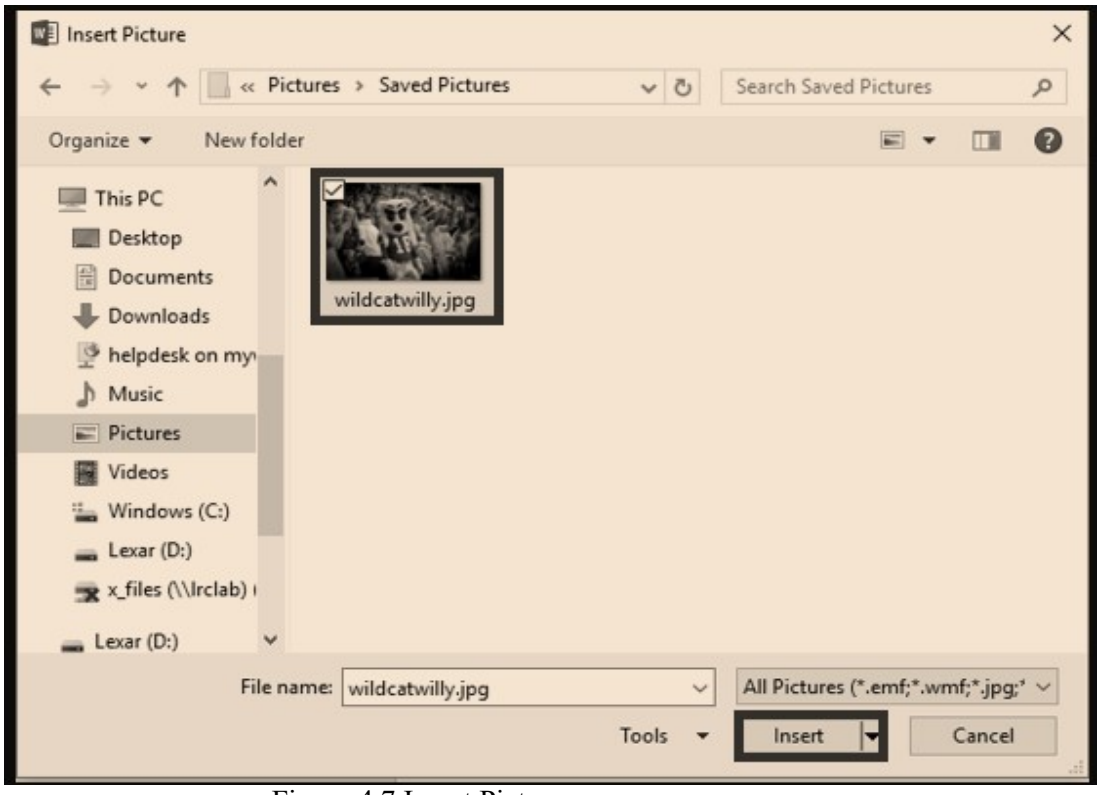


Figure 4.7 Insert Picture

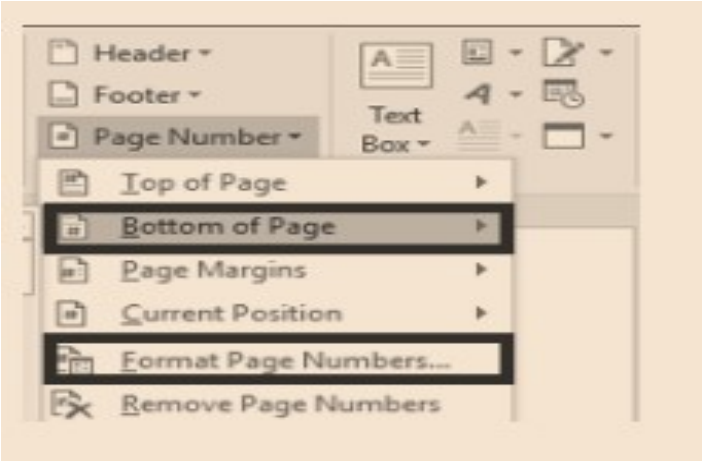


Figure 4.8 Insert Page Number

These features improve communication, especially in academic and professional reports.

4.2.3 Saving a Document

Saving your work frequently is crucial to prevent data loss. Microsoft Word offers multiple save options:

- **Save**: Stores the document in its current state.
- **Save As**: Saves a new copy of the document, allowing users to rename the file or change its location or format.
- **File Formats**:
  - .docx – default Word format
  - .pdf – for sharing and printing
  - .rtf, .txt, .html – for specific compatibility needs

Cloud Saving through OneDrive also allows automatic backups and online access from multiple devices.

4.2.4 Reviewing and Proofreading

The **Review** tab includes essential tools for ensuring document quality:

- **Spelling and Grammar Check**: Highlights errors and suggests corrections.
- **Thesaurus**: Offers synonyms to improve word choice.
- **Word Count**: Displays the number of words, characters, and pages.
- **Track Changes**: Allows multiple users to suggest edits and comments in collaborative documents.
- **Comments**: Add feedback or notes without changing the original content.

These tools help create polished and accurate documents.

4.2.5 Printing and Sharing Documents

After completing the document, users may want to print or share it:

- **Printing**:
  - Navigate to File > Print
  - Select the printer, number of copies, and page range
  - Preview the layout before printing

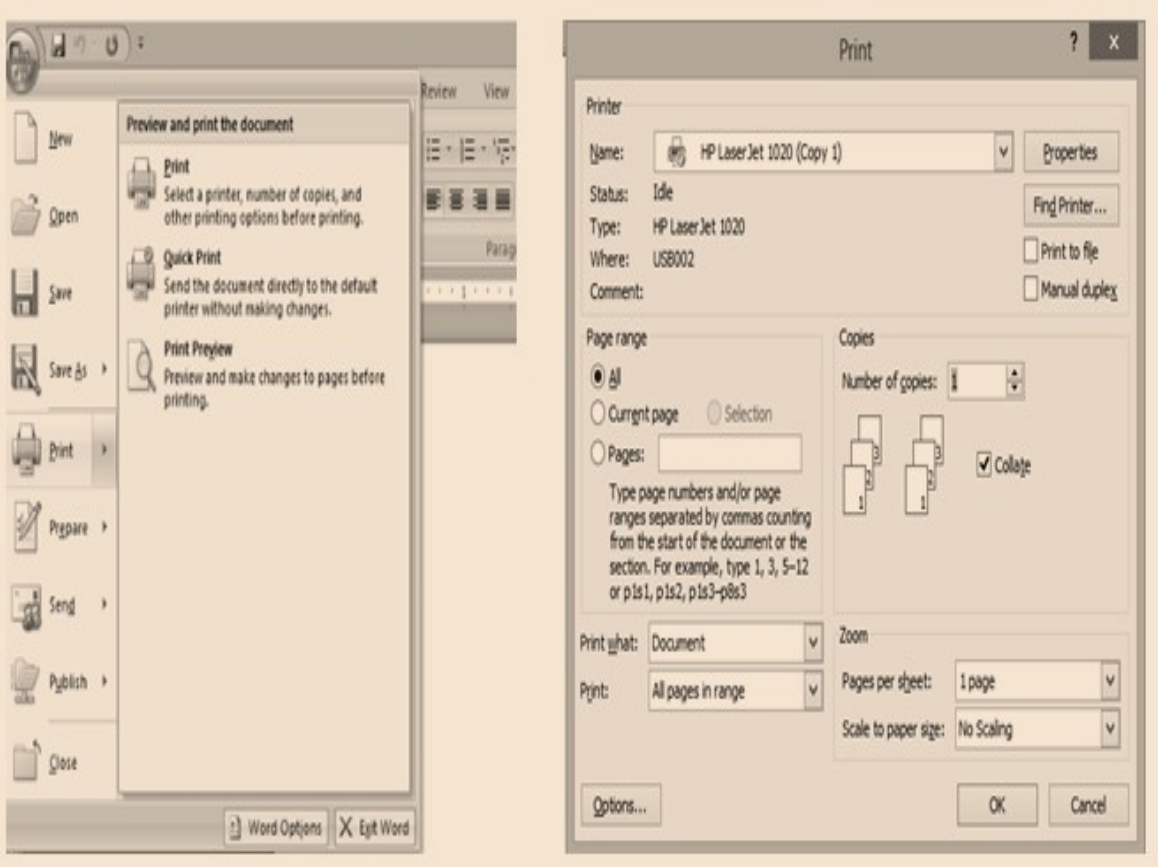


Figure 4.9 Printing a document

- **Sharing**:
  - Save as PDF and email as an attachment
  - Use File > Share to send via OneDrive
  - Export to different formats or embed into websites

Microsoft Word also supports password protection and document encryption for secure sharing.

4.3 FORMATTING:

Formatting in Microsoft Word refers to the process of arranging and styling text and other elements to improve the appearance, readability, and structure of a document. Proper formatting ensures that a document looks professional, follows required standards, and effectively communicates information to its audience.

Microsoft Word provides a wide range of formatting options that can be applied to individual characters, words, paragraphs, sections, or entire pages.

4.3.1 Types of Formatting in MS Word

There are several categories of formatting in Microsoft Word, which can be broadly classified into:

- **Text (Character) Formatting**
- **Paragraph Formatting**
- **Page Formatting**
- **Document Formatting (Using Styles and Themes)**

Each category contains specific tools and commands designed to enhance the content visually and functionally.

4.3.2 Text (Character) Formatting

Text or character formatting refers to changes made to the appearance of individual letters, words, or sentences. These options are located under the **Home** tab in the **Font** group.

Common Text Formatting Options:

Formatting Tool	Description
Font Style	Changes the type of text (e.g., Arial, Calibri, Times New Roman)
Font Size	Adjusts the size of text (e.g., 10pt, 12pt, 14pt)
Bold	Makes text darker and thicker for emphasis
Italic	Slants the text to the right
Underline	Adds a line beneath the text
Text Color	Changes the color of selected text
Highlighting	Applies a background color to highlight specific text
Case Change	Converts selected text to uppercase, lowercase, or other cases
Strikethrough	Draws a line through the middle of the text
Subscript/Superscript	Positions text slightly below (subscript) or above (superscript) the normal line

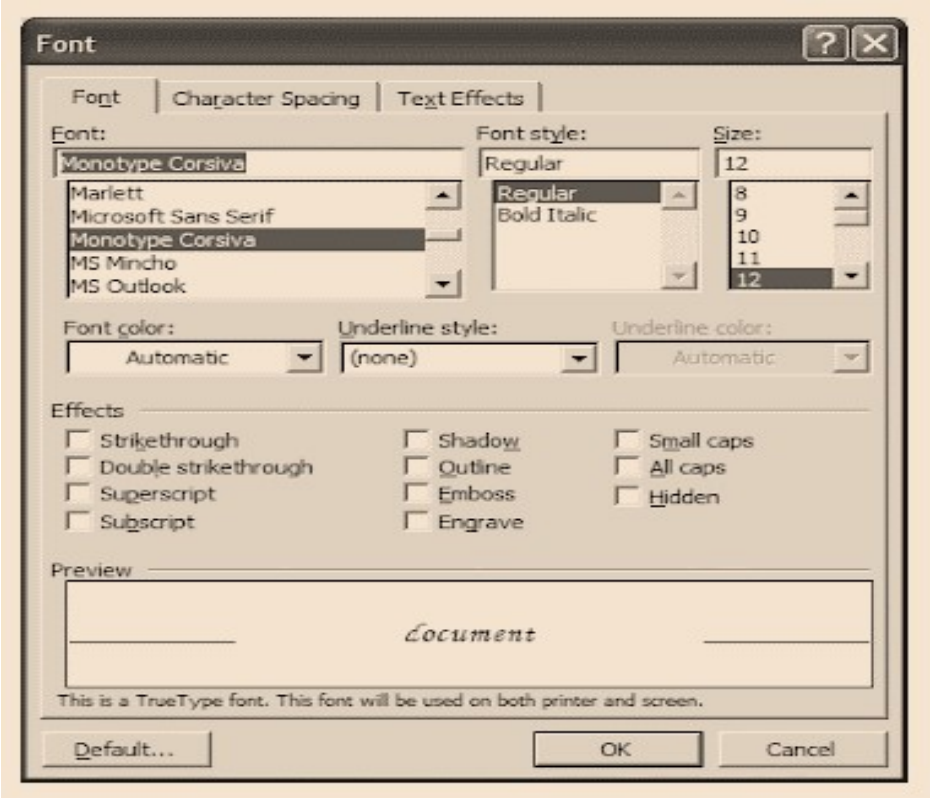


Figure 4.10 Text Formatting

These options allow users to emphasize important parts of their content and maintain visual consistency throughout the document.

4.3.3 Paragraph Formatting

Paragraph formatting refers to the alignment, spacing, indentation, and layout of text blocks. These options are found under the Home tab in the Paragraph group.

Key Paragraph Formatting Features:

- **Alignment**:
  - **Left**: Text aligns with the left margin (default).
  - **Center**: Centers the text in the middle of the page.
  - **Right**: Aligns text to the right margin.
  - **Justify**: Distributes text evenly between both margins.
- **Line and Paragraph Spacing**:
  - Adjusts the amount of vertical space between lines or between paragraphs.
  - Common line spacing: Single, 1.5 lines, Double.
- **Indentation**:
  - Moves text inward from the left or right margin.
  - **First Line Indent**: Only the first line of the paragraph is indented.
  - **Hanging Indent**: All lines except the first line are indented.
- **Bullets and Numbering**:
  - Adds bullet points or numbered lists to organize information.
- **Borders and Shading**:
  - Adds outlines and background colors to paragraphs for emphasis.

Proper paragraph formatting improves readability and helps structure long documents logically.

4.3.4 Page Formatting

Page formatting involves the overall layout of the document, such as margins, orientation, size, and columns. These options are located in the **Layout** tab.

Important Page Formatting Options:

Option	Description
Margins	Sets the distance between text and the edge of the paper (e.g., Normal, Narrow, Wide)
Orientation	Chooses between Portrait (vertical) and Landscape (horizontal) layout
Page Size	Selects paper size (e.g., A4, Letter, Legal)
Columns	Splits text into two or more columns (common in newsletters)
Breaks	Adds section or page breaks to separate content
Line Numbers	Adds numbers to each line (useful in legal documents)
Hyphenation	Automatically breaks words between lines using hyphens

Effective page formatting ensures that the document prints correctly and adheres to presentation or submission standards.

4.3.5 Styles and Themes

Styles:

A style is a set of formatting instructions applied to text to ensure consistency. Common styles include Normal, Heading 1, Heading 2, Title, and Subtitle. Users can customize or create their own styles.

Applying styles helps with:

- Maintaining a uniform look across headings and body text
- Automatically generating a Table of Contents
- Quick formatting changes throughout the document

Themes:

Themes are collections of fonts, colors, and effects that give a cohesive appearance to the entire document. They are especially useful in professional or academic documents and are found under the Design tab.

4.3.6 Formatting Tools and Shortcuts

Microsoft Word provides several tools to enhance formatting efficiency:

- **Format Painter**: Allows users to copy formatting from one part of the text and apply it elsewhere.
- **Clear Formatting**: Removes all formatting from selected text and returns it to default.
- **Shortcut Keys**:
  - **Ctrl + B**: Bold
  - **Ctrl + I**: Italic
  - **Ctrl + U**: Underline
  - **Ctrl + E**: Center alignment
  - **Ctrl + L**: Left alignment
  - **Ctrl + J**: Justify
  - **Ctrl + 1 / 2 / 5**: Line spacing (single, double, 1.5)

Using these shortcuts improves productivity and speeds up the document formatting process.

4.3.7 Best Practices in Formatting

- Use consistent font styles and sizes throughout the document.
- Avoid using too many colors or font types.
- Apply heading styles for better structure and easy navigation.
- Align text properly to maintain clean margins.
- Use bullet and numbered lists to present information clearly.
- Leave appropriate spacing between sections.
- Preview the document before printing to ensure the layout is correct.

Formatting is a crucial component of document creation in Microsoft Word. It enhances both the appearance and effectiveness of written content. Mastering text, paragraph, page, and style formatting enables users to create documents that are clear, professional, and visually appealing. Whether working on academic assignments, office reports, or creative writing, proper formatting significantly improves the quality and impact of the final product.

4.4 EDITING:

Editingin Microsoft Word refers to the process of reviewing, modifying, and refining the content of a document to improve its clarity, accuracy, and presentation. It involves making changes to the text, correcting errors, reorganizing content, and enhancing readability. Editing is an essential step in document preparation, especially for formal writing such as reports, letters, academic assignments, and business documents.

Microsoft Word provides a wide range of tools to support both basic and advanced editing tasks, making the process efficient and user-friendly.

4.4.1 Basic Text Editing Functions

Microsoft Word offers several basic tools for editing text. These tools are commonly found under the Home tab and are essential for making quick changes to the content.

Common Editing Functions:

- **Cut (Ctrl + X):** Removes selected text or content from one place.
- **Copy (Ctrl + C):** Copies the selected content without removing it.
- **Paste (Ctrl + V):** Inserts the cut or copied content at the desired location.
- **Undo (Ctrl + Z):** Reverses the last action or step.

- **Redo (Ctrl + Y):** Repeats the last undone action.
- **Delete:** Removes selected characters, words, or elements.

These basic commands allow users to efficiently move and organize content within a document.

4.4.2 Selecting Text

Before editing, text must be selected. This can be done in the following ways:

- **Using the Mouse:** Click and drag to highlight text.
- **Double-Click:** Selects a word.
- **Triple-Click:** Selects an entire paragraph.
- **Ctrl + A:** Selects the entire document.

Proper selection enables precise editing, formatting, and replacement of content.

4.4.3 Find and Replace

The Find and Replace tool is useful for quickly locating specific words or phrases and replacing them throughout the document.

How to Use:

- Go to the Home tab >Editing group> Click Replace.
- Enter the word to be found in the "Find what" box.
- Enter the replacement word in the "Replace with" box.
- Click Find Next, Replace, or Replace All as needed.

This tool is helpful for correcting repeated mistakes, updating terms, or rephrasing content consistently.

4.4.4 Inserting and Deleting Text

Inserting Text:

- Click where you want to add text and start typing.
- Use the Enter key to start a new paragraph.
- Use Tab to insert indentation.

Deleting Text:

- Use Backspace to delete characters to the left of the cursor.
- Use Delete to remove characters to the right of the cursor.
- Select and press Delete to remove larger sections.

These basic functions are used frequently during content modification.

4.4.5 Using the Clipboard

The Clipboard temporarily stores copied or cut text and allows users to paste it elsewhere. Word also has a Clipboard Panel that stores up to 24 copied items.

To Use:

- Cut or copy content.
- Open the Clipboard Panel under the Home tab to see recent items.
- Click any item in the panel to paste it into the document.

This feature is especially useful when managing content from multiple parts of a document.

4.4.6 Spell Check and Grammar Tools

Microsoft Word includes powerful tools to help detect and correct spelling and grammar errors:

Automatic Underlining:

- Red: Spelling error
- Blue: Grammar/syntax error
- Green (older versions): Contextual or formatting issues

Accessing the Editor:

- Go to Review tab > Click Spelling & Grammar or Editor
- Review and accept or ignore suggested corrections
- Use the Thesaurus to find synonyms for better word choice

These tools help improve writing quality and ensure linguistic accuracy.

4.4.7 Track Changes and Comments

The Track Changes feature is useful for collaborative editing. It allows users to see edits made by others, accept or reject changes, and keep track of revisions.

To Use:

- Go to Review tab > Track Changes
- All insertions, deletions, and formatting changes will be highlighted
- Use the Accept or Reject buttons to finalize edits

Adding Comments:

- Select text and click New Comment
- Add notes or feedback in the margin without changing the content

This is especially useful for teamwork, peer review, or academic editing.

4.4.8 Editing with Navigation Pane

The Navigation Pane provides a way to quickly move through sections of a document, especially when headings are used.

To Open:

- Go to View tab> Check Navigation Pane

Users can click on headings to jump to different parts, search for text, and reorganize content more efficiently.

4.4.9 Advanced Editing Tools

MS Word also includes advanced tools for editing large or complex documents:

- **Compare Documents**: Compare two versions of a document to see differences.
- **Restrict Editing**: Control who can make changes to a document.
- **Protect Document**: Add passwords to prevent unauthorized editing.
- **AutoCorrect**: Automatically fixes common typing errors as you type.

These tools add an extra layer of control and precision during the editing process.

Best Practices for Editing

- Always save a backup before making major edits.
- Use Track Changes for collaborative or formal editing.
- Read the document aloud to catch unnoticed errors.
- Check formatting consistency (fonts, spacing, alignment).
- Use the Find tool to locate overused words or repeated mistakes.

Editing in Microsoft Word is a vital step in document preparation. It involves more than just correcting errors—it includes reorganizing, refining, and enhancing content to meet the desired purpose and audience expectations. By mastering the various editing tools in Word, users can produce polished, accurate, and high-quality documents with ease and efficiency.

4.5 SUMMARY:

Microsoft Word is a popular program used to create and edit documents for school, work, or personal use. It can be opened from the Start menu, desktop, or taskbar. Users can start a new document, use a template, or open a saved file. Word makes it easy to type and edit text, change font style, size, color, and alignment, and format paragraphs with spacing, bullets, and numbering.

Page layout options let users set margins, paper size, and add columns or breaks. Pictures, tables, charts, links, headers, footers, and page numbers can also be added. Documents can be saved in formats like .docx or .pdf and stored on OneDrive. Word includes tools to check spelling, grammar, word count, and allows comments or track changes. It also lets users print or share documents easily. Features like styles, themes, cut, copy, paste, and find/replace help in editing, while AutoCorrect and document protection help keep the work neat and safe.

4.6 KEY TERMS:

Formatting, Editing, Templates, Track Changes, Styles

4.7 SELF ASSESSMENT QUESTIONS:

1. What are the steps to create a new document in Microsoft Word?
2. How can you format a paragraph using alignment and spacing options?
3. Which tool would you use to replace a word throughout the entire document?
4. What is the purpose of using Track Changes and Comments in Word?
5. How can you save a document in PDF format using Microsoft Word?

4.8 SUGGESTIVE READINGS:

1. Microsoft Word 365 Step by Step by Joan Lambert – Microsoft Press.
2. Word 2019 For Dummies by Dan Gookin – Wiley.
3. Official Microsoft Support Articles: https://support.microsoft.com/word

Dr. G.Neelima

LESSON- 5

MS WORD - MAIL MERGE, TABLES, AND INDEX PRINTING

OBJECTIVES:

After going through this unit, you should be able to:

- Use mail merge, and
- Create and modify date in tables
- Understanding Indexing

STRUCTURE:

5.1 Introduction to Mail Merge

5.1.1 Select Type of Mail Merge

5.1.2 Select Your Recipients

5.1.3 Edit Recipients List

5.1.4 Inserting Fields

5.1.5 Previewing Merge Document

5.1.6 Finish Merge Document

5.2 Tables

5.2.1Create a Table

5.2.2 Modify Table Structure

5.3 Indexing

5.5 Summary

5.6 Key Terms

5.7 Self-Assessment Questions

5.8 Suggestive Readings

5.1 MAIL MERGE:

Mail merge technique allows you to create a document (such as form letter, invitation) and set of information of recipients (such as names, addresses) and you merge them. When you merge the form letter with list of names, you can generate the required letter for each of the recipient. In this procedure, you have to create a letter once, combining it with your address list and send to multiple recipients.

For example, if you want to invite 50 people on workshop. You will write just one letter of invitation (known as the main document, explained ahead), you will store all the 50 names and addresses of people whom you intend

to invite in one place (known as data source, explained ahead), then you will tell word to merge the two information (the main document and the data source) and create 50 letters of invitation each with different name and address on it.

To perform mail merge, you will need the following:

- A main document
- Data Source or recipient list

Main Document
This is the identical content that appears in each letter that mail merge creates.The main document can be a form letter, labels, envelopes, an invitation or any type of document. The document can be appended with some unique information and field names such as name, address which MS Word will replace automatically with the recipient list or data source. The main idea behind this technique is that you create one document but that can be sent to people in a recipient group.

Data Source
A data source is a file that contains the name, addresses list that varies in each copy of merged document. The data source may be created during the merge process using templates or an existing file. The existing file may be Excel (.xlsx) file. Mail Merge uses this data to make different letters with the same content but different personal information.

The following example is explained with built-in templates for data file. Excel are discussed in Unit 11 and 12, therefore, after going through these units, you may try to create data file using this application and use it with mail merge using the following similar steps.

Perform the following six steps for using mail merge:

1. Select Type of Mail Merge
2. Select Your Recipients
3. Edit Recipients List
4. Inserting Fields
5. Previewing Merge Document
6. Finish Merge Document

5.1.1 Select Type of Mail Merge

To use mail merge first you must create a letter using the following steps:

- Select **Mailings** tab on Ribbon
- Click on '**Start Mail Merge**'option, a drop down menu will display a list of options. You can select any option such as Letters, Labels, Email messages, Directories, Envelopes or step by step mail merge wizard. Suppose, you have selected letters as an option (see figure 5.1) and type the sample letter for your practical work (see figure 5.2)

Fig. 5.1: Select type of Mail Merge

Fig. 5.2: Sample Letter

5 .1.2 Select Your Recipients

Now, you should create the recipient list using the following steps:

You can click on '**Select Recipients**' button from the 'Start Mail Merge'group of **Mailings tab** ribbon. (see Figure 5.3)

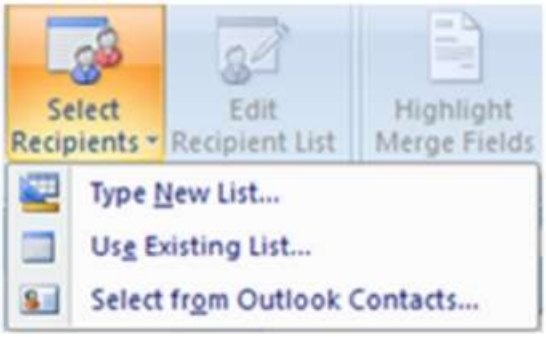


Fig. 5.3: Select Recipients menu

A list appears, in which you have to select '**Type New List'**. The following screen will be display for data entry

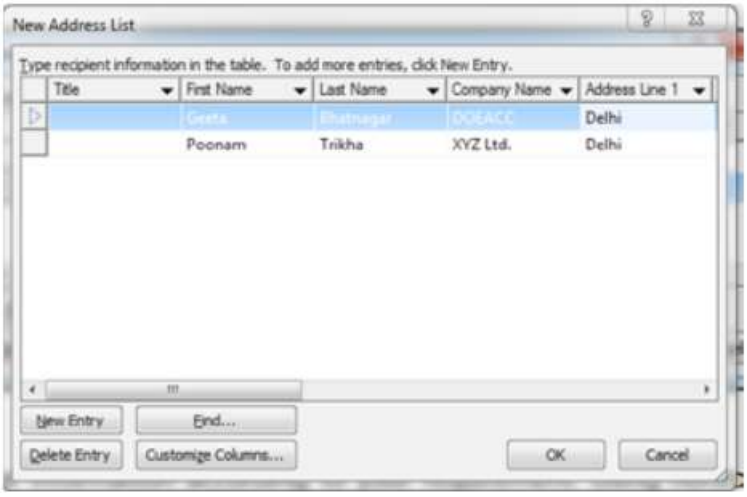


Fig. 5.4: Recipients Data Entry window

- Type the information according to your requirement. Using horizontal scroll bar, you can see long list of field names

To add more data row in the table, click on New Entry button (see Figure 5.4).

- To save your data entry, click on OK button, it will open a dialog box, in which give a name to your data and select '**Save**' button (see Figure 5.5)

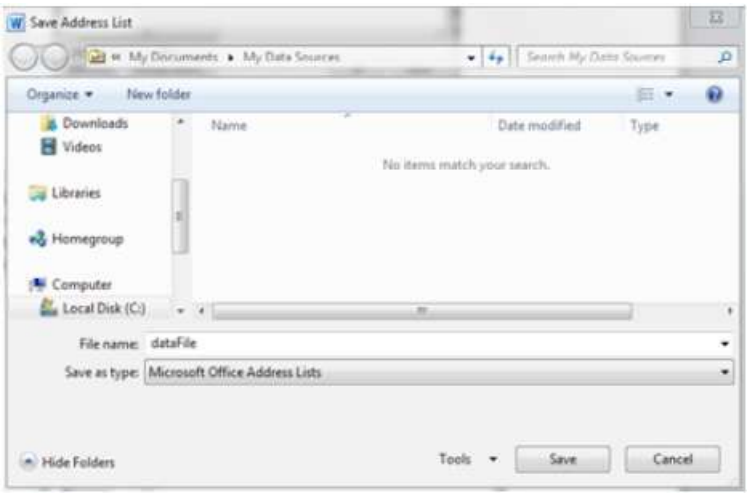


Fig. 5.5: Save Address List

5.1.3 Edit Recipients List

When you build up a data list, 'Edit Recipient List' will be enabled. If you select this button, then it will allow you to make changes in the existing recipients list. You can find '**Edit Recipient List'** option on '**Start Mail Merge'** group under Mailings tab. Using the following window, you can do the following with data:

- Select or deselect your data from the list
- Sort your data items as ascending or descending order
- Add or change your data list
- Filter data in the list (see Figure 5.6)

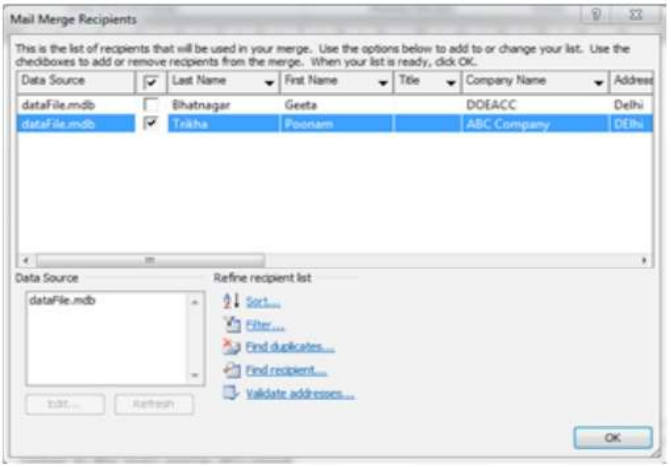


Fig. 5.6: Edit Mail Merge Recipient List Screen

5.1.4 Inserting Fields

In previous section, you have created recipient list. Now, it is time to insert fields into a main document. Although, Word provides you a variety of options for adding merge field names to the main document.

To insert merge field names,

- Click on **Mailings tab** Ribbon.
- Select '**Insert Merge Field'** option from '**Write & Insert Fields'** group.
- A menu list appears, in which you have to select fields name (for example, First_name) see Fig. 5.7
- Repeat the above step until your desired fields are added to the merge document and add fields like the highlighted portion of the following Fig. 5 .8

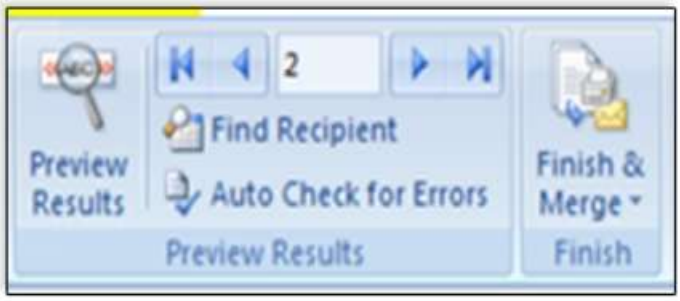


Fig. 5.7: Merge Document

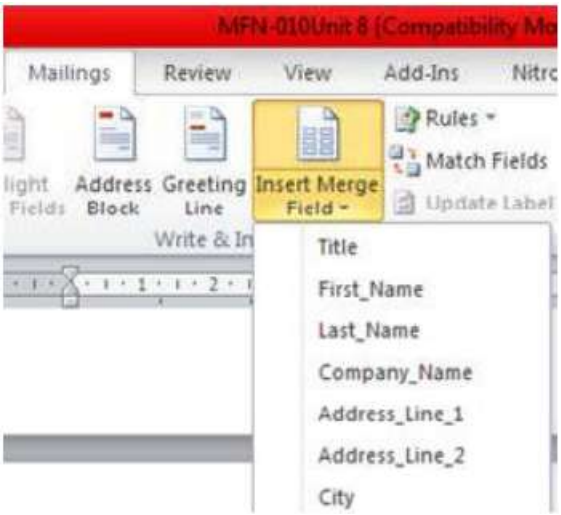


Fig. 5.8: Inserted Mail Merge Field Name

5.1.5 Previewing Merge Document

Using MS Word '**Preview Results**' option, you can see your results of merge document.

To previewing result data,

- Click on '**Preview Result'** from Preview Result group on Mailings tab Ribbon (see Fig. 5.9)

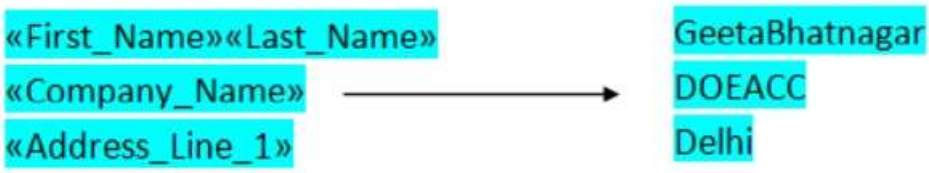


Fig. 5.9: Preview Result Screen

Fig. 5.10: Mail Merge data Display

Here, you have seen in the Fig. 1.9, the record pointer is on number 2. So, it will show record no. 2 in place of fields' syntax (see figure –1.10). You can see rest record using record pointer

5.1.6 Finish Merge Document

In the previous sections, you have created a letter, inserted recipient list and previewed your merged document. Now, you can give a finishing touch to your mail merge document and to create a separate document for each record in recipient list and send them all directly to the printer or email.

For this, you can use any one option from the following:
1) Edit Individual Documents (creating a single copy for each recipient)
2) Print a document
3) Send E-Mail Messages

Edit Individual Documents

If you want to edit individual documents, you can complete your mail merge process by the following ways:

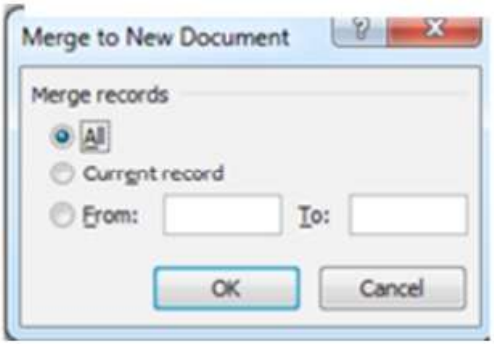


Fig. 5.11: Merge document options

- Click the '**Finish & Merge'** button on Finish group from Mailings tab ribbon.
- From menu, select Edit Individual Documents. A window will display in which you can select 'All' option and click on Ok button (see Fig. 5.11)
- A new window will open for each recipient.

In similar way, you can use other two options such as **Print documents** and **send via mail messages** for your merging document. For mail merge document printing, Printer is necessary for printing separate copy and to send through email but for this process, outlook setting and Internet connection is necessary

5.2 TABLES:

A table is an arrangement of information in rows and columns. The intersection of rows and columns is called cells. The 'cell' content is the text, which can be formatted with the same methods as are used to format any other text. Each table has a border associated with it, which appears as soon as the table is created but it is not binding on you, you can delete the border if you want.

Tables are used for various tasks such as presenting text information and numerical data in a table format. In this section, you will learn how to create or draw a table, convert text to a table; apply table styles, insert rows and columns, split and merging cells, as well as import Excel sheet.

5.2.1 Create a Table

Fig. 5.12: Create a Table

For creating a blank table, you can do the following steps:

- Place your insertion point in the document where you want to place the table.
- Select the Insert tab of the ribbon (see Fig. 5.12).
- Click on table button under Tables group, you can create a table one of the five ways:

1. Drag your mouse over the palette to create a table with the highlighted number of rows and columns **OR**
2. Click on **Insert table** and enter required number of rows and columns **OR**
3. Click on the '**Draw table'**; create your table by using pencil object **OR**
4. Click on '**Excel Spreadsheet'** to import excel table **OR**
5. Click on '**Quick Tables'** option to create built-in tables such as Calendar, tabular list and matrix form table structure.

Once you create a table, place the cursor in the cell where you wish to enter the information and begin typing.

Using Insert Table option to create table

If you choose the 'Insert Table' option of Tables Group of Insert tab to create the table, dialog box is invoked as given below with table size and AutoFit behaviour options. You can specify the table values for the following way

Fig. 5.13: Define table behavior

- In **table size**, you can specify the number of columns and rows which you want to create.

- In **AutoFit behaviour**, you can select one option, out of the three options which are given below:

- **Fixed column width:** Using this option, you can select a specific width for the columns you create

- **AutoFit to contents:** This option automatically adjusts the width of columns to fit the table data you enter.

- **AutoFit to window:** This option resizes the table so that it fits the contents within a Web browser window.

5.1.2 Modify Table Structure

For modification of the table, you can **add new rows and columns**; **delete existing rows and columns** as well as table, **split and merge cells**. In designing part of the table, you can give a new look to table which contains the color and design of border, rows, columns, header and footer.

When you want to modify table, click on the table and you will have notice that there are two new tabs on the Ribbon: **Design** and **Layout.** These tabs include all the features that are mentioned above.

To modify a table, click on table that you have created and then click the **Layout** Tab on the Ribbon. This Layout tab allows you to:

- Insert Rows and Columns
- Split or Merge Cells
- Delete table, Rows and/or Column

Fig. 5.14: Layout tab for modifying a table

Insert Rows in Existing Table:

For the following way, you can insert row either above or below of the selected row in existing table:

- Place the insertion point in the row where you want to insert the new row.
- To insert one row at a time, click on '**Insert Above'** option on 'Rows & Columns' Group of Layout tab Ribbon for above of the selected row.
- In similar way, click on '**Insert Below'** to insert one row below to your selection.

- Alternatively, right click on the table and select Insert (see Fig. 1.15) and click on "Insert Rows Above", if you want to add new row above your selected row. Otherwise, you can click on "Insert Rows Below". This option will add new row below your selection.
- This way, you can add one row at a time. **Insert Columns in Existing Table:**

You can add new columns to the left or right of the existing column. Do the ollowing:

1. Place the insertion point in the column where you want to insert the new column.

2. To insert new column left or right to your selection, click on '**Insert Left'** or '**Insert Right'** option on 'Rows & Columns' Group of Layout tab Ribbon (see Fig. 5.14).

3. Alternatively, right click on the table and select Insert from shortcut menu (see Fig. 1.15) and click on "Insert column to the Left" to insert column left. Otherwise, you can click on "Insert column to the Right" to insert column right

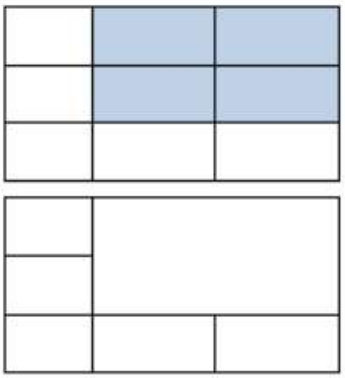


Fig. 5.15: Shortcut menu for inserting rows and columns in existing table

Split and Merge Cells

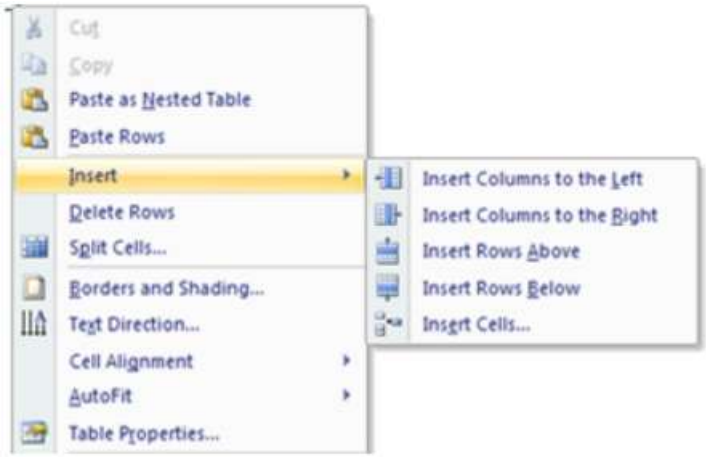


Fig. 5.16: Merging Cells

When working with tables, you may want to merge two or more cells or split cells into more cells.

Merging Cells:

The 'Merge Cells' feature is used to combine the content of multiple cells into one cell.

- Select the cells to be merge
- Right click within the selected cells
- Select Merge Cells,
- The selected cells are merged (figure 5.16).

Splitting Cells

You can split one cell at a time. To Split the cells, by using the following steps:

- Select the cell to be split
- Right click on selected cell; a split cells dialog box will appears.(see Fig.5 .17).
- In split cells dialog box, select the desired number of columns and rows.
- Click ok and selected cells are split according to your requirement. You can see all the splitting cell process in the following figure- 5.17

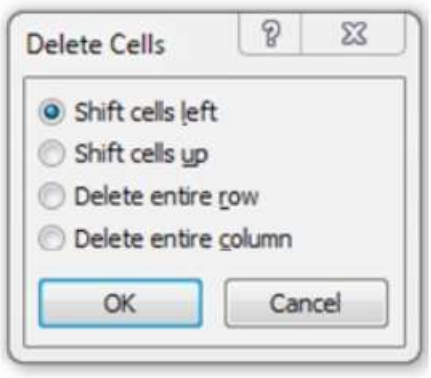


Fig. 5.17: Split Cells Dialog Box

Delete table, cells, Rows and/or Column

Word provides you four options (see Fig. 1.18) for deletion as:

1) Delete Cells

2) Delete Columns

3) Delete Rows

4) Delete Table

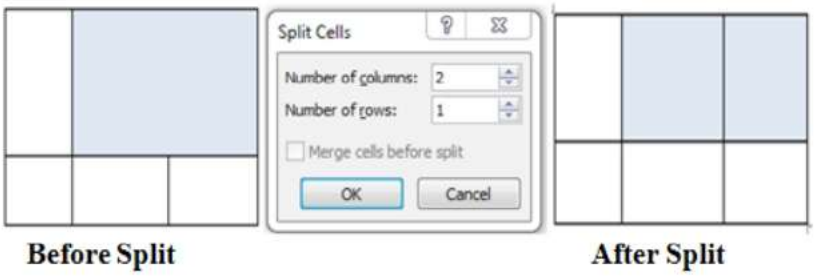


Fig. 5.18: Delete options for Table

1) **To Delete cells**:

- When you select Delete Cells option, a Delete Cells dialog box gets invoked as displayed in the Fig. 5.19.

- Select one of the following options in the Delete Cells dialog box

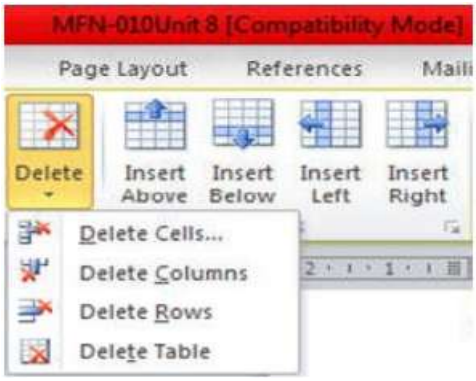


Fig. 5.19: Delete Cells dialog box

- Select the **Shift cells left** radio button to move the cells in the row to the left of the deletion.

- Select the **Shift cells up** radio button to move the cells in the column up after the deletion.

- Select the **Delete entire row** radio button to delete the row that contains the selected cell.

- Select the **Delete entire column** radio button to delete the column that contains the selected cell.

2) Delete Columns:

- Place the cursor on the column which you want to delete and select **Delete columns** to delete the particular column (figure 5.18).

3) Delete Rows:

- Place the cursor on the row which you want to delete and select **Delete rows** to delete the particular row (figure 5.18).

4) Delete Table:

- Select the table for deletion and click on '**Delete Table'** option to delete the entire table (figure 5.18).

5.3 INDEXING:

In a well-structured Index you would use the Mark button more frequently than the Mark All button to create an Index.

Generally there are three tasks in creating an Index:

1. Review the document and plan for Index entries.

2. Mark the content for the Index.

3. Generate the Index.

As mentioned, the purpose of this tutorial based on the limited time available, will simply use keywords instead of concepts to illustrate how to mark and generate different types of Indexes from basic to complex.

This is NOT the correct way to create a well-structured Index!

Marking Content
The strategy I use to mark content for an Index is to keep my final version of a document clean and create a copy of the file that then becomes my final working copy. I do this once everything in the document is set and it is ready to publish.

Why do I do this?

There is a chapter on removing the Index entries for documents you are revising, however, I keep forgetting what that process is so find it just easier to keep a clean copy of the document without Index entries just in case I need it. I usually put the word "final" in the filename of the document with the Index entries and save it to a folder called "Final Version" just to keep things straight.

I've also found that when I use that process to remove Index field entries, it does not remove any sub-entry or cross-reference text and I have to do that manually for each instance of the text that is left This can take a lot of time..

The first thing to do is to review the document and decide which topics are going to be in the Index and if you need to create sub-entries or cross references.

The next step is to create the individual Index entries by marking them in the document.

You can add the Index entries as you write, however, I find it easier to review the document once I've finished it and then mark the Index entries. The choice is yours.

I also find it easier to select the word or phrase I want added to an Index before I launch the Mark Entry dialog. I am using a screen reader so if you are not using adaptive technology,

you may or may not find this helpful. Because I am using a screen reader, I also close the Mark Entry after marking each entry so that focus doesn't accidentally go to it and mess up my document.

If you do select the word or phrase, it appears in the Mark Entry dialog and you can either keep it as the Index entry or replace it. As someone using adaptive technology, this confirms to me that this is the place I want to create the Index entry field for use in the Index.

The Index tools are on the References Ribbon. The keyboard command to open the Mark Index Entry dialog is Alt + S, N.
You can also press Alt + Shift + X to open the Mark Entry dialog.

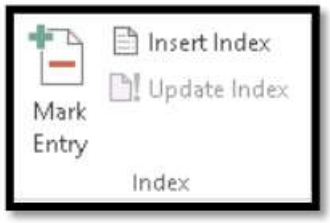


Figure 5.20 Index Group on the References Ribbon

When the Mark Entry dialog opens, your focus is in the edit field labeled Main Entry. This has the word or phrase you selected in it.

For the following image, the word I selected in my document was the word Video with a capital V.

I referred to switches at the start of this document. One strategy to avoid going in and messing about with switches is to make a change in this field. The word I selected for the following image has a capital letter. Other words I select may not. I simply edit the words to have capital letters when I want capital letters and this is transferred to my Index when I create it.

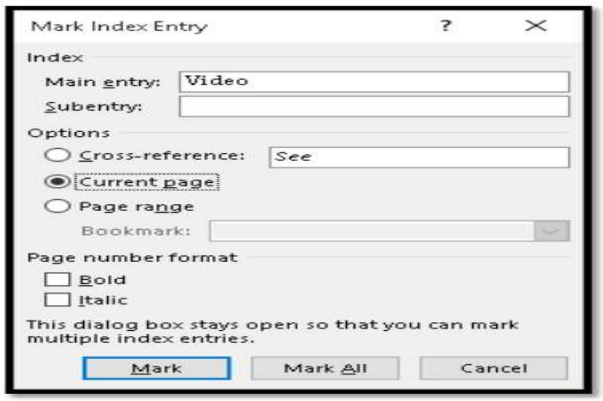


Figure 5.21 Mark Index Entry

If you want to go back to the Mark Entry field in the Mark Entry dialog, press Alt + E.

You can Tab or press Alt + S to move to the Subentry field. This lets you identify a place in the Index where topics might overlap. For example, Cover page might have a sub-entry of Page Layout or Page Layout might have a sub-entry of Cover Page.

You don't have to enter a sub-entry unless there is one. You can also create multiple levels of sub-entries by using a semi-colon (;) to separate terms. We'll look at these more complex Indexes later in this tutorial.

The next choice to make is a series of three radio buttons to help define what will be displayed in the Index.

- Press Tab or Alt + C to move to the first radio button for the Cross-reference field.

The Cross-reference field has an edit box associated with it which will take focus when you make this choice. This is where you can say, for example, see Multimedia. In our example, video might be a multimedia topic.

- Press the Down Arrow or Alt + P to move to the Current Page radio button. This is

the default choice and will use the current page to gather Index entries.

- Press the Down Arrow or Alt + N to move to the Page Range radio button. When you

choose this option, the Bookmarks edit field is shown and takes focus. This is one way to create multiple Indexes, but we'll get into that later!

The last two settings in the Mark Index Entry dialog are to make the page number format bold or italic. Both are check boxes and are unchecked by default. As with your regular document content, keep in mind that a page with all page numbers in Italics or Bold may cause readability barriers for people with disabilities.

Press Alt + B to move to and check the Bold check box. You can use the Spacebar to uncheck this if you are exploring the dialog and don't want to apply this attribute.

Press Alt + letter I to move to and check the check box to make the page number format Italic. You can use the Spacebar to uncheck this if you are exploring the Mark Index Entry dialog and don't want to apply this attribute.

The last choice you need to make is to choose whether you only want this instance of the word or phrase in the Index or whether you want Word to add all instances of the word to the Index.

If you've chosen to use a page range for this entry, Selecting Mark All will only mark entries within that page range. Use this when creating multiple Indexes in a document where you only want certain words for each chapter to appear in the Index. If you want all instances of the word, for example "video" to be available for all Indexes, don't use this setting.

Most of the time you will use the Mark button (Alt + M). You probably won't want EVERY instance of a word or phrase marked for the Index. Keep in mind that adding these fields

increases the size of your document. This is one reason that, as a document author, you should review and analyze your finished document to decide which words and phrases are to be included in an Index.

For our samples, and for that sense of instant gratification, we'll choose Mark All (Alt + A)

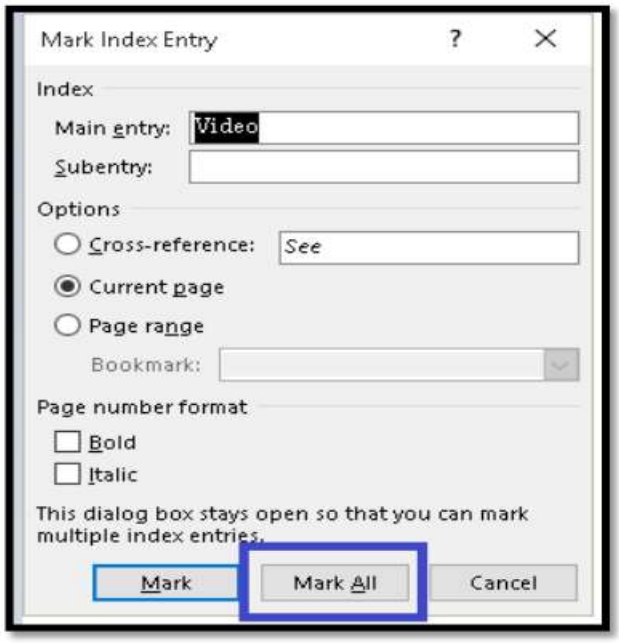


Figure 5.22 Mark Index Entry showing Mark All button.

Once an Index entry is added to a document, the Paragraph or Formatting Marks appear so that you can see the Index code. The Index code comes immediately after the word or phrase you selected (or where your cursor point was in the document when you marked the entry).

What you will see is something like: {XE: “Video”}
If you are using a screen reader, both the content and all of the Paragraph Marks will be read to you including spaces between words. This might sound confusing as you move through the text to find other words or phrases to add to your Index

Figure 5.23 Show/Hide formatting and hidden text.

If you see the code for the Index entry in your document, press Alt + H, 8 to turn off the Paragraph Marks option. This is a toggle and will either turn on or off the Paragraph Marks. You can also click on the Paragraph Formatting button in the Home Ribbon, Paragraph Group to turn on or off the Paragraph Formatting.

The following image shows a paragraph in the sample document where the word “video” has been marked for entry in the Index. The codes are visible just after the word

Generating an Index
Once you mark the words and phrases that you want included in your Index, it’s time to generate the Index and review your work!

Position your cursor where you want the Index to appear.

I usually set up a page at the end of the document, after any appendices, for the Index. In my documents, the Index is the very last item in the document.

Once your cursor is where you want the Index to appear, press Alt + S, X which opens the Index dialog

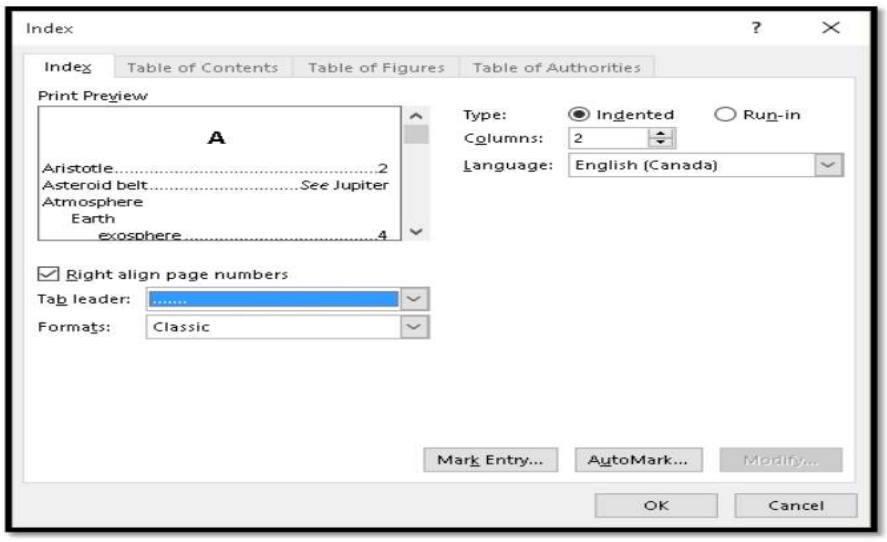


Figure 5.24 Index dialog for generating an Index.

The Index dialog looks a bit like the Table of Contents dialog in that on the left is a sample of what your Index is going to look like and you have options to add Dot Leaders or use one of the existing templates for your Index.

5.5 SUMMARY:

Mail Merge
Mail Merge allows you to create personalized versions of documents by combining a main document with a data source.

Key accomplishments:
- Creating form letters, envelopes, labels, or email messages for mass distribution
- Connecting and using data from various sources (Excel, Access, Outlook)
- Inserting merge fields to personalize documents
- Using rules and filters to target specific recipients
- Creating and saving merge templates for future use

Indexing
Word's indexing capabilities help create professional navigation aids for longer documents.

Key accomplishments:
- Marking index entries (main entries and subentries)
- Creating cross-references between index entries
- Generating and formatting a comprehensive index
- Updating indexes as document content changes
- Creating custom index styles and formatting

Tables
Tables organize information in rows and columns for better readability and data

5.6 KEY TERMS:

Main Document, Data Source, Merge Fields , Recipient List: rules, Merge to Printer, Merge to Email ,Mail Merge Wizard, Field Codes ,Index Entry

5.7 SELF ASSESSMENT QUESTIONS:

- Can you identify the five main types of documents that can be created using Mail Merge?
- How do you create a data source within Word if you don't already have one?
- What are the steps to connect an existing Excel spreadsheet as a data source?
- How do you insert merge fields into your document?
- What is the process for marking a simple index entry in Word?
- How do you create a subentry in an index?
- Can you explain how to mark an index entry that spans multiple pages?
- What is the procedure for creating a cross-reference in an index?
- How many different ways can you create a table in Word?
- Can you convert existing text into a table? What are the requirements?
- How do you add or delete rows and columns in a table?
- What is the process for merging and splitting cells?

5.8 SUGGESTIVE READINGS:

Books and Official Resources

1. **"Microsoft Word In Depth" by Faithe Wempen**
   - Comprehensive coverage of advanced Word features including detailed chapters on mail merge, tables, and document organization
2. **"Microsoft Word Step by Step" by Joan Lambert**
   - Practical, exercise-based approach with dedicated sections on mail merge and tables
3. **Microsoft's Official Word Documentation**
   - Free, continually updated resource directly from Microsoft with detailed tutorials
4. **"Special Edition Using Microsoft Office Word" by Faithe Wempen**
   - In-depth coverage of advanced features with troubleshooting tips
5. **"Microsoft Word Secrets: The Why and How of Getting Word to Do What You Want" by Herb Tyson**
   - Focused on advanced features and workarounds for complex document challenges

Dr D.V. Chandrashekar

LESSON- 6

MS POWERPOINT: SLIDE CREATION, DESIGN, AND ANIMATION

OBJECTIVES:

1. What is MSPower point 2010 and what are its functions
2. What are the components of Powerpoint 2010
3. The tools used in MS Power point 2010 and their use

STRUCTURE:

6.1 Introduction to Power Point

6.1.1 MS-Powerpoint Slide Creation

6.2 Features of MS PowerPoint

6.2.1 Insert – Clipart, Video, Audio, etc.

6.2.2 Slide Design

6.2.3 Animations

6.3 Uses of PowerPoint Presentation

6.3.1 Use of templates

6.3.2 How to create slides in Powerpoint

6.3.3 Add and format text

6.4 How to create animations in PowerPoint (Windows)

6.4.1 How To Add Animations to PowerPoint In 2 Simple Steps

6.4.2 What is a slide master

6.4.3 Action buttons

6.5 Summary

6.6 Key Terms

6.7 Self-Assessment Questions

6.8 Suggestive Readings

6.1 INTRODUCTION:

Microsoft PowerPoint is a presentation software within the Microsoft Office suite that allows users to create dynamic and visually appealing slide shows for various purposes, including business presentations, educational lectures, and professional conferences.
Here's a more detailed introduction:

Functionality:

- **Creating Presentations:**

PowerPoint enables users to design and format presentations using slides, which can contain text, images, graphics, charts, videos, and animations.

- **Visual Communication:**

It's designed to enhance communication by providing a visual aid to accompany oral presentations, making information more engaging and easier to understand.

- **Versatility:**

PowerPoint is a versatile tool used in various contexts, including business meetings, educational presentations, marketing materials, and even personal projects.

Key Features:

- **Templates and Themes:**

PowerPoint offers a wide variety of pre-designed templates and themes to streamline the design process and ensure a professional look.

- **Slide Layouts:**

Users can choose from various slide layouts to organize content effectively, such as title slides, content slides with bullet points, and image slides.

- **Multimedia Integration:**

PowerPoint allows users to easily insert images, videos, audio, and other multimedia elements to enhance the presentation.

- **Animations and Transitions:**

Users can add animations and transitions to slides to make the presentation more dynamic and engaging.

- **Presenting and Sharing:**

PowerPoint provides features for presenting slideshows, exporting them to different formats, and sharing them with others.

6.1.1 MS-Powerpoint Slide Creation

PowerPoint (PPT) is a powerful, easy-to-use presentation graphics software program that allows you to create professional-looking electronic slide shows.

The image given below shows the main page of MS PowerPoint, where a person lands when the program is opened on a computer system:

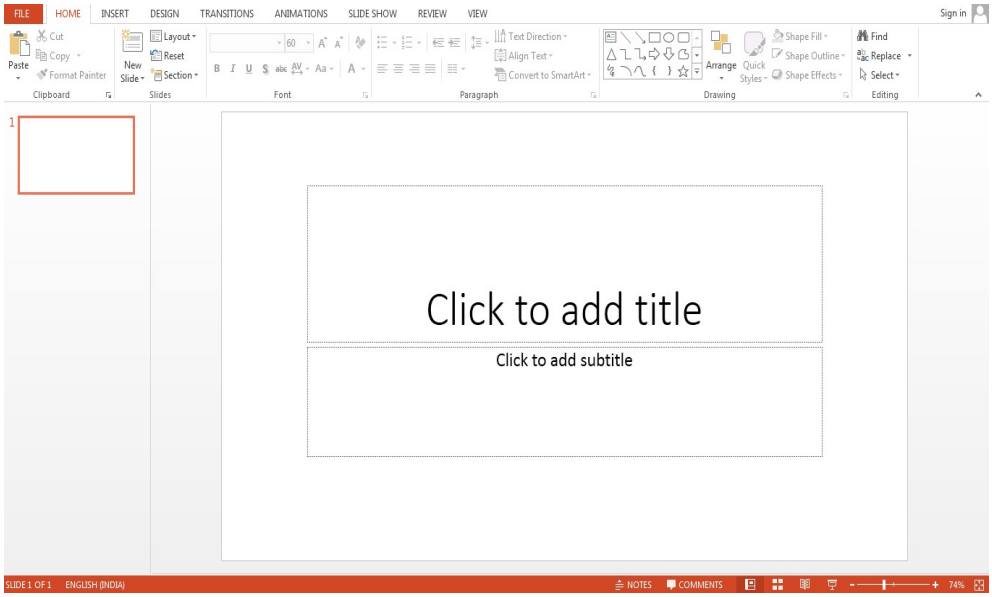


Figure 6.1 Basic Structure of Power Point Slide

Follow the steps below to open MS PowerPoint on a personal computer:
1. Click on the start button
2. Then choose “All Programs”
3. Next step is to select “MS Office”
4. Under MS Office, click on the “MS PowerPoint”

A blank presentation is open on the screen. According to the requirement, a person can modify the template for a presentation and start using the program.
The following elements can be added to a Power point slide:
1. Clip Art
2. Graphs
3. Tables
4. Photographs
5. Charts
6. Media Clips
7. Videos

All these elements are mainly used to enhance presentation skills and make the slide more interactive.

6.2 FEATURES OF MS POWERPOINT:

There are multiple features that are available in MS PowerPoint which can customise and optimise a presentation. The same have been discussed below.

- **Slide Layout**

Multiple options and layouts are available based on which a presentation can be created. This option is available under the “Home” section and one can select from the multiple layout options provided.

The image below shows the different slide layout options which are available for use:

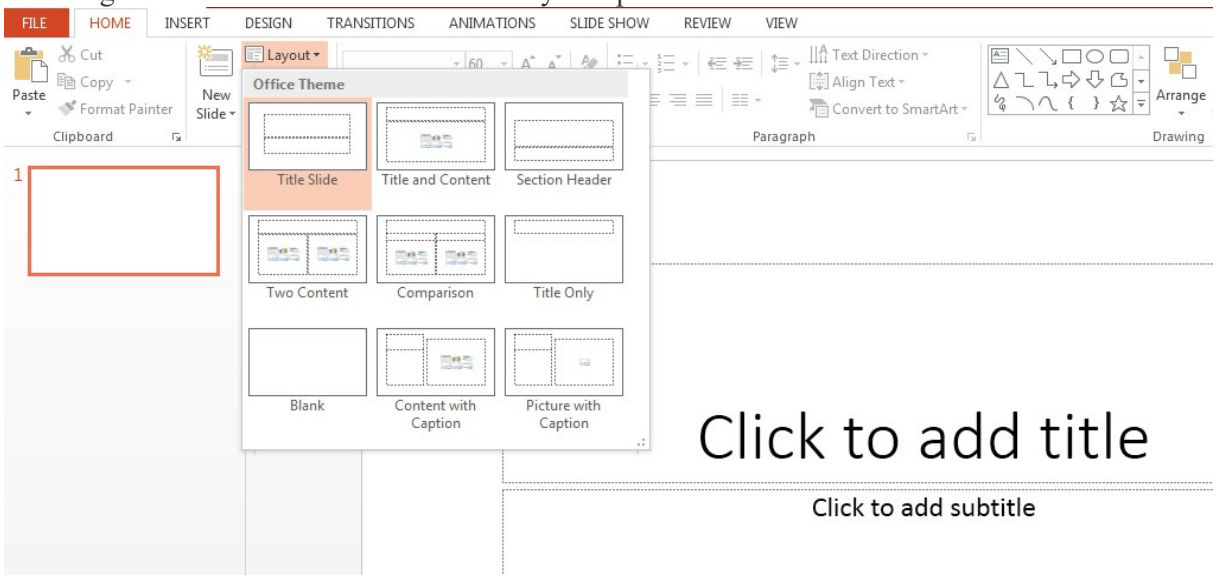


Figure 6.2 Overview of Power Point Menu

1.2.1 Insert – Clipart, Video, Audio, etc.

Under the "Insert" category, multiple options are available where one can choose what feature they want to insert in their presentation. This may include images, audio, video, header, footer, symbols, shapes, etc.

The image below shows the features which can be inserted:

Figure 6.3 Overview of Insert Menu

6.2.2 Slide Design

MS PowerPoint has various themes using which background colour and designs or textures can be added to a slide. This makes the presentation more colourful and attracts the attention of the people looking at it.

This feature can be added using the "Design" category mentioned on the homepage of MS PowerPoint. Although there are existing design templates available, in case someone wants to add some new texture or colour, the option to customise the design is also available. Apart from this, slide designs can also be downloaded online.

Refer to the below for slide design:

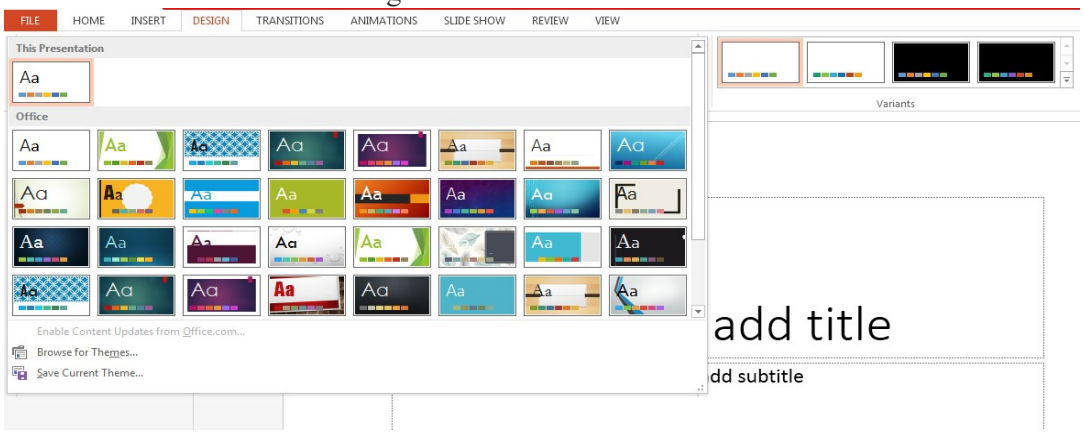


Figure 6.4 Slide Customization

6.2.3 Animations

During the slide show, the slides appear on the screen one after the other. In case, one wants to add some animations to the way in which a slide presents itself, they can refer to the "Animations" category.

The different animation styles available on PowerPoint are:

Figure 6.5 Customize Animation Menu

Apart from all these options; font size, font style, font colour, word art, date and time, etc. can also be added to a PPT.

6.3 USES OF POWERPOINT PRESENTATION:

PowerPoint presentations are useful for both personal and professional usage. Given below are a few of the major fields where PPT is extremely useful:

- **Education –** With e-learning and smart classes being chosen as a common mode of education today, PowerPoint presentations can help in making education more interactive and attract students towards the modified version of studying
- **Marketing –** In the field of marketing, PowerPoint presentations can be extremely important. Using graphs and charts, numbers can be shown more evidently and clearly which may be ignored by the viewer if being read
- **Business –** To invite investors or to show the increase or decrease in profits, MS PowerPoint can be used
- **Creating Resumes –** Digital resumes can be formed using MS PowerPoint. Different patterns, photograph, etc. can be added to the resume
- **Depicting Growth** – Since both graphics and text can be added in a presentation, depicting the growth of a company, business, student's marks, etc. is easier using PPT

6.3.1 Use of templates

1. In the left pane, select **New**.
2. Select an option:

- To create a presentation from scratch, select **Blank Presentation**.
- To use a prepared design, select one of the templates.
- To see tips for using PowerPoint, select **Take a Tour**, and then select **Create**, .

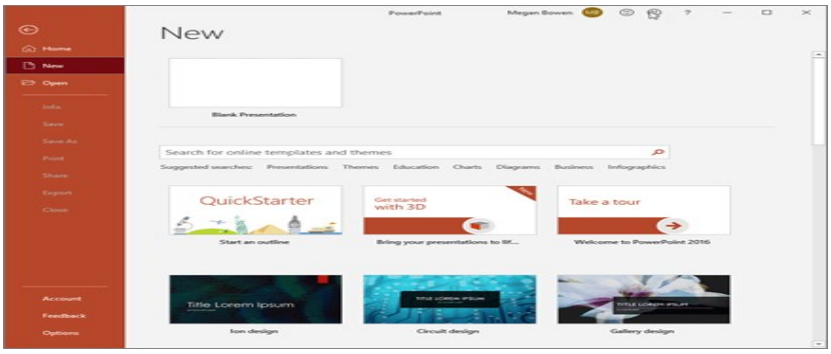


Figure 6.6 Slide Templates

1.3.2 How to create slides in power point

Add a slide

1. In the thumbnails on the left pane, select the slide you want your new slide to follow.
2. In the **Home** tab, in the **Slides** section, select **New Slide**.
3. In the **Slides** section, select **Layout**, and then select the layout you want from the menu.

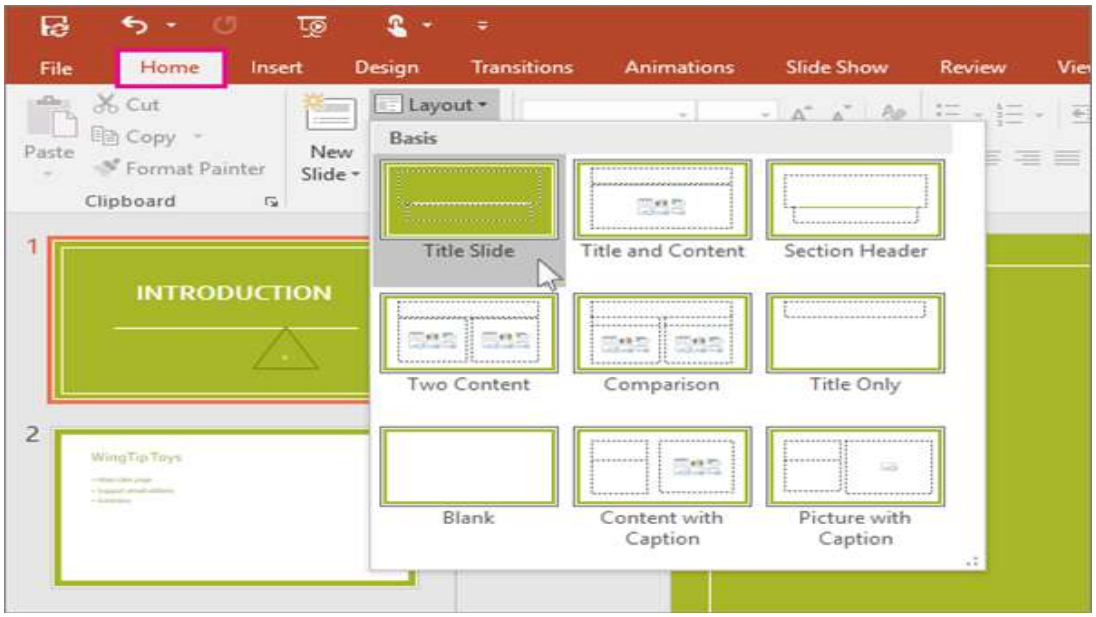


Figure 6.7 Layout Slide of Power Point

1.3.3 Add and format text

1. Place the cursor inside a text box, and then type something.
2. Select the text, and then select one or more options from the **Font** section of the **Home** tab, such as **Font**, **Increase Font Size**, **Decrease Font Size**, **Bold**, **Italic**, **Underline**, etc.
3. To create bulleted or numbered lists, select the text, and then select **Bullets** or **Numbering**.

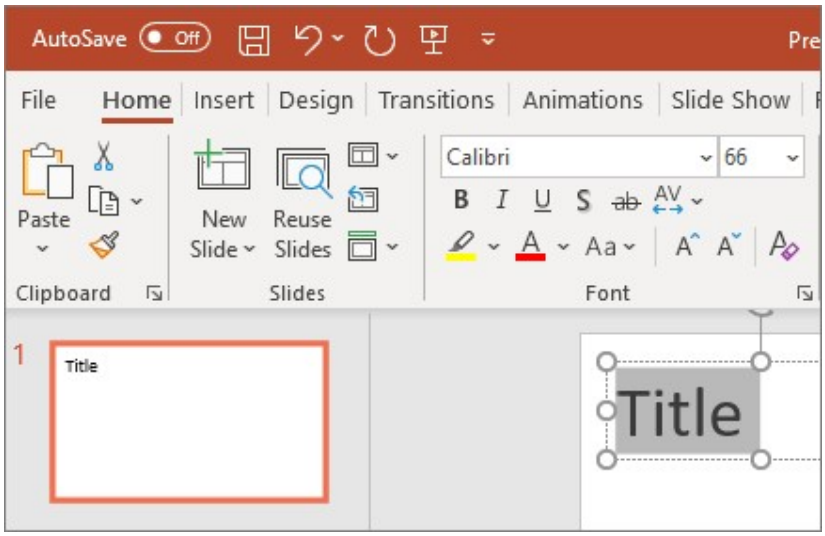


Figure 6.8 Insert Menu options

Add a picture, shape, and more

1. Go to the **Insert** tab.
2. To add a picture:

- In the **Images** section, select **Pictures**.
- In the **Insert Picture From** menu, select the source you want.
- Browse for the picture you want, select it, and then select **Insert**.

To add illustrations:

- In the **Illustrations** section, select **Shapes**, **Icons**, **3D Models**, **SmartArt**, or **Chart**.
- In the dialog box that opens when you click one of the illustration types, select the item you want and follow the prompts to insert it.

6.4 HOW TO CREATE ANIMATIONS IN POWERPOINT (WINDOWS):

Animation are visual effects which make your texts, images, shapes, or charts come 'alive.' They catch your audience's attention and helps them engage with you and your presentation. You first to have to **click on the object you want to animate**, then **click on the Animations tab**.

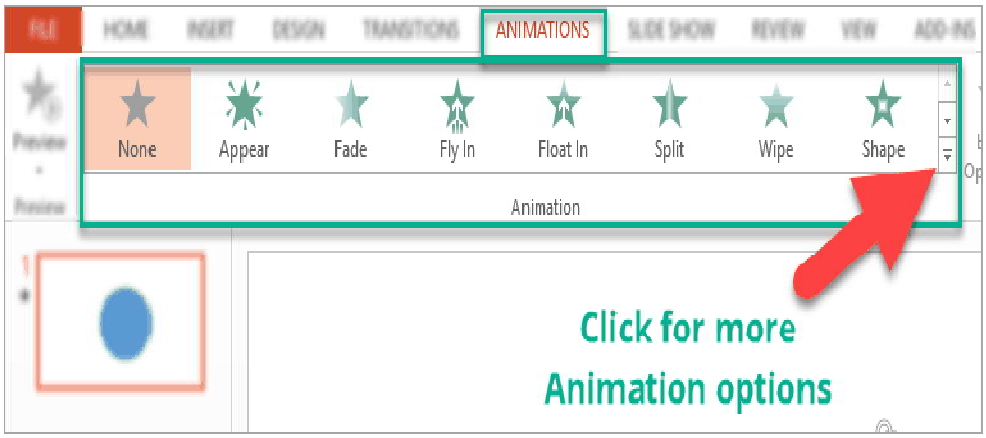


Figure 6.9 The Animations panel in PowerPoin)

The default view shows 8 animations, but if you click on the little arrow like you see on the screenshot above, then you'll see many more options like you see here:

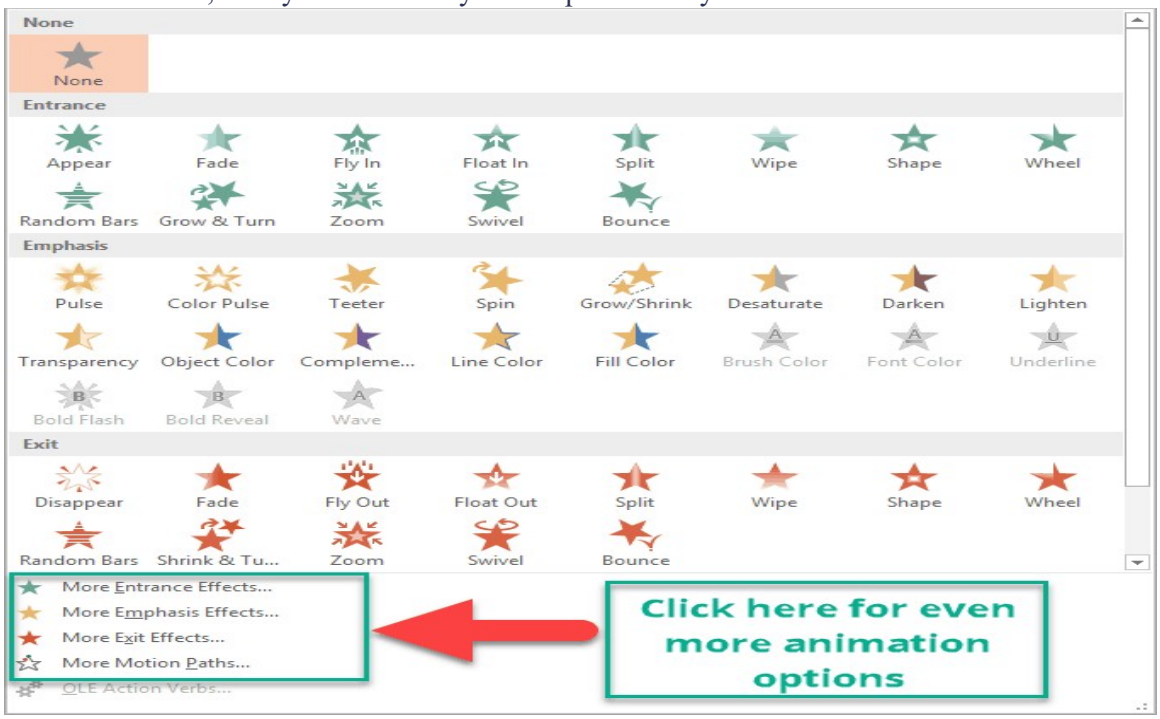


Figure 6.10 More Animation options in PowerPoint – choose from Entrance, Exit,

Emphasis and Motion Paths animations

You can choose from the following types of animations:

- **Entrance animations.** The icons for this type of animation are **colored green**. If you use this type of animation for an object, that object will **enter the slide** according to your timing preferences.
- **Exit animations.** The icons for this type of animation are **colored red**. The object you've added this animation to will **exit the slide**.
- **Emphasis animations.** The icons for this kind of animation are **colored yellow**. The object won't enter or exit the slide (unless you've added that particular animation to the same object) but will emphasize and draw attention to that object.
- **Motion path animations.** This type of animation will allow you to **move an object from one spot to another.** You can specify or draw the path you want the object to take.

In the next sections, we'll go over how to add the *4 different types of animations* to your slides.

6.4.1 How To Add Animations to PowerPoint In 2 Simple Steps

To add entrance animations to any object in PowerPoint, here's what you need to do:

- Click on the object you want to animate. In this example, we will use a blue smiley face.

Figure 6.11 The blue smiley face we will be animating for this tutorial

2. Select the animation you want to use. For this example, we used the **entrance animation** – **fade** animation. We set the duration to 2.75 seconds. You will notice in the screenshot below that the fade animation has no effect options (the button is greyed out).

Figure 6.12 The different options available for Fade animation in PowerPoint)

Here's what those numbers in the screenshot mean:
1 – We chose the Fade animation.
2 – There are no Effect Options for Fade Animation (the button is grayed out and not clickable)
3 – You can adjust settings via the Advanced Animation, Timing, and Animation Pane sections
4 – The number (1) refers to the fact that we've added 1 animation to our object (this is the Fade animation).
5 – You can click on the Preview button to preview the animation.

The next tab is **Timing,** You can choose when you want your animation to start (on click, with previous or after previous), the duration of the animation, add a delay and choose whether you want the animation to repeat.

Figure 6.13 Animation Time Menu

6.4.2 What is a slide master?
When you want all your slides to contain the same fonts and images (such as logos), you can make those changes in one place—the Slide Master, and they'll be applied to all your slides.
To open Slide Master view, on the **View** tab, select **Slide Master**:

Figure 6.14 Slide View Setting Menu

The master slide is the top slide in the thumbnail pane on the left side of the window. The related layout masters appear just below the slide master (as in this picture from PowerPoint for macOS):

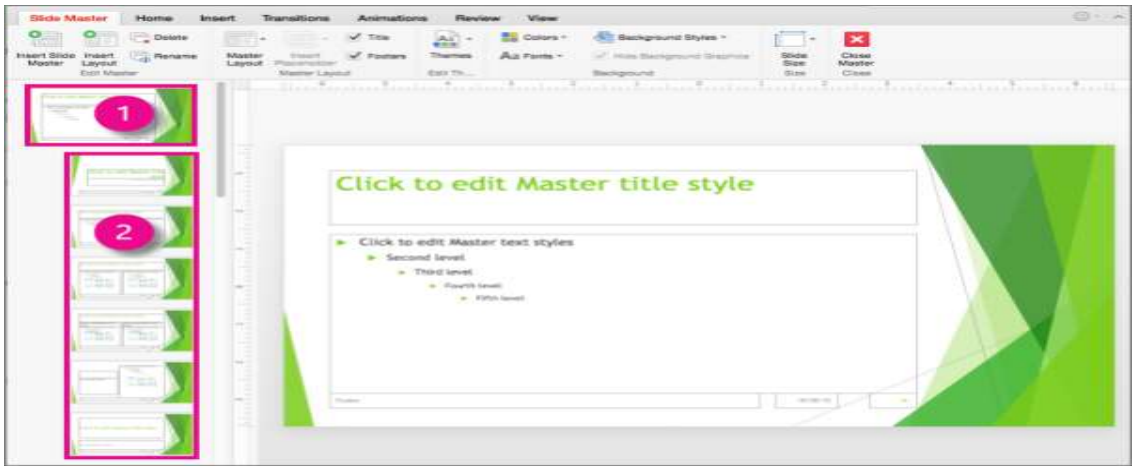


Figure 6.15 Slide View Setting Menu

1 Slide master
2 Layout masters

When you edit the slide master, all slides that are based on that master will contain those changes. However, the majority of changes that you make will most likely be to the layout related to the master.

When you make changes to layout masters and the slide master in Slide Master view, other people working in your presentation (in Normal view) can't accidentally delete or edit what you've done. Conversely, if you're working in Normal view and find that you're unable to edit an element on a slide (such as, "why can't I remove this picture?") it may be because the thing you're trying to change is defined on the slide master or a layout master. To edit that thing, you must switch to Slide Master view.

6.4.3 Action buttons

Use an Action button to do an action in your presentation, such as going to the next or a specific slide, running an app, or playing a video clip. There are two basic steps for setting up an Action button:

1. First, you pick an action button from the Shapes gallery
2. Then you assign an action that occurs when (during Slide Show) you click the shape or point the mouse at it.
3. The idea is that when you deliver your presentation, you can click or point at an action button to:
4. Go to the next slide, the previous slide, the first slide, the last slide, the most recent slide viewed, a specific slide number, a different PowerPoint presentation, or a Web page.
5. Run a program

Put an action button on your slide

Some examples of built-in action button shapes in the Shapes gallery include right and left arrows (commonly understood symbols for going to next, previous, first, and last slides), for playing videos or sound bites, and more.

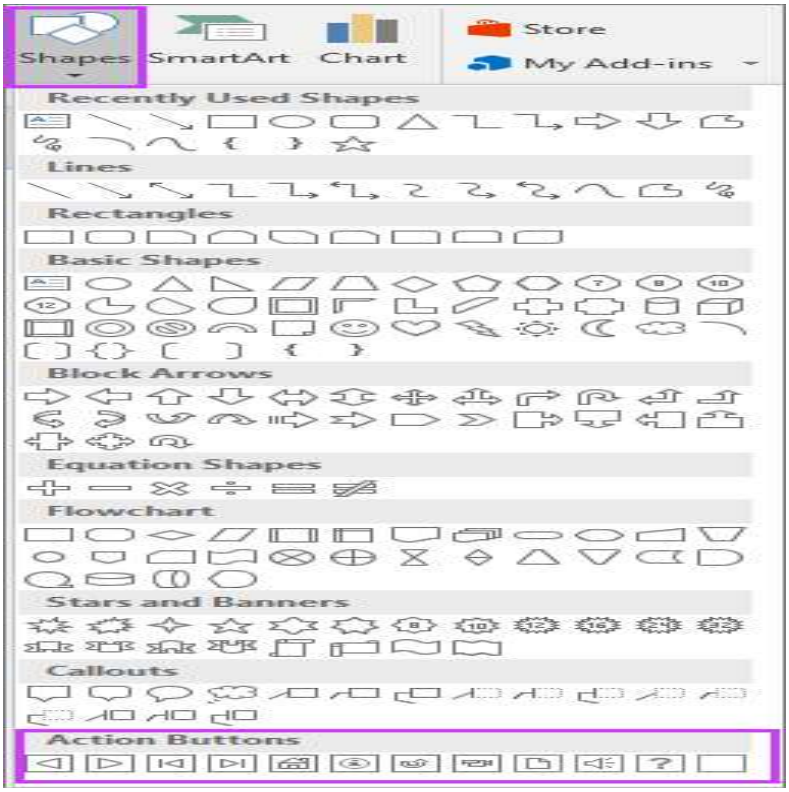


Figure 6.16 Showing Auto Shapes and Action Button Menu

Shows the Shapes menu on the ribbon in PowerPoint with Action Buttons highlighted

On the Insert tab, click Shapes, and then under Action Buttons at the bottom of the menu, click the button shape that you want to add.

Click a location on the slide, and then drag to draw the shape for the button.

Keep the dialog box open and go to the next procedure.

Assign an action

If you want the action to occur when the action button is simply pointed at, do the following steps on the Mouse tab of the dialog box.

To choose the action that takes place when you click or move the pointer over the action button, do one of the following:

Choose Hyperlink to and then select the destination (for example, the next slide, the previous slide, the last slide, or another PowerPoint presentation) that you want the hyperlink action to go to.

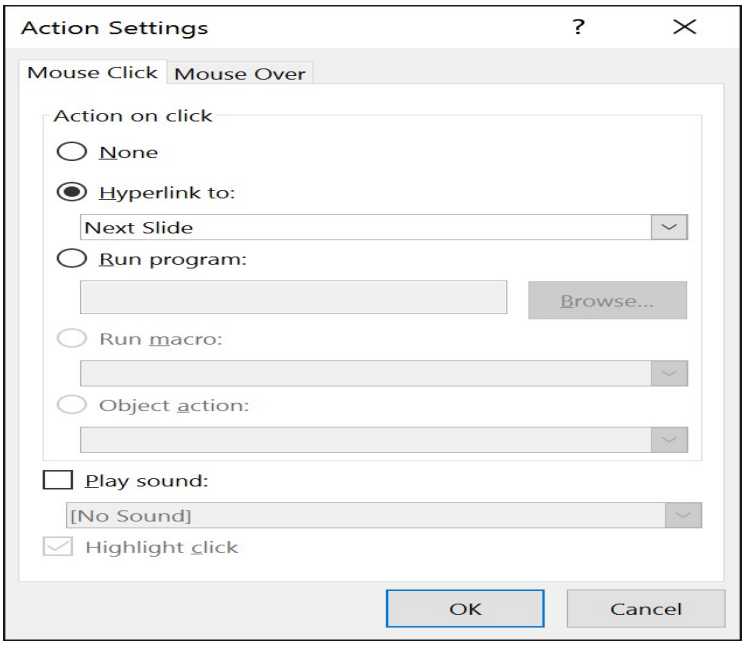


Figure 6.17 Action Setting Menu

When you've finished choosing your action, select OK.

6.5 SUMMARY:

- Creating and designing professional presentations
- Working with slides, templates, and themes
- Adding and formatting text, images, charts, and multimedia
- Creating animations and transitions
- Using master slides for consistent design
- Delivering effective presentations
- Creating custom templates and themes
- Using SmartArt for visual representations

6.6 KEY TERMS:

1. Slide, Slide Deck ,Slide Master, Layout, Theme ,Template,Placeholder ,Text Box ,SmartArt ,Chart, Table, WordArt, ShapeAnimationTransition ,Animation Panel Timing, Trigger

6.7 SELF-ASSESSMENT QUESTIONS:

1. Can you create a new presentation from scratch and from templates?
2. Do you know how to add, delete, duplicate, and reorder slides?
3. Can you apply different slide layouts appropriate for different content?
4. Are you able to insert and format text in placeholders and text boxes?
5. Do you know how to check spelling in your presentation?
6. Can you add speaker notes to your slides

6.8 SUGGESTIVE READINGS:

1. **"PowerPoint 2019 For Dummies" by Doug Lowe**
   - Perfect for beginners with clear explanations and step-by-step instructions
2. **"Presentation Zen: Simple Ideas on Presentation Design and Delivery" by Garr Reynolds**
   - Focuses on design principles and effective visual communication
3. **"Slide:ology: The Art and Science of Creating Great Presentations" by Nancy Duarte**
   - Excellent resource on visual storytelling and presentation design
4. **"Talk Like TED: The 9 Public-Speaking Secrets of the World's Top Minds" by Carmine Gallo**
   - Combines PowerPoint best practices with effective presentation delivery techniques
5. **"Better PowerPoint: Quick Fixes Based On How Your Audience Thinks" by Stephen M. Kosslyn**
   - Cognitive psychology approach to creating effective slides
6. **Online Resources**
7. **Microsoft's Official PowerPoint Training Center**
   - Free tutorials directly from Microsoft covering basic to advanced features
8. **LinkedIn Learning (formerly Lynda.com) PowerPoint Courses**
   - Video-based instruction with practical exercises and example files
9. **PowerPoint Creative Blog by Microsoft**
   - Latest updates, tips, and creative inspiration
10. **"PowerPoint Magic" course by Echo Swinford**
    - Advanced techniques for PowerPoint power users
11. **PowerPoint templates and design resources from Envato Elements or SlideModel**
    - Professional templates to study and adapt

Dr . D.V. Chandrashekar

LESSON- 7

PRESENTATION ENHANCEMENTS AND SLIDESHOW FEATURES STRUCTURE

OBJECTIVES:

After going through this unit, you should be able to:

- create a PowerPoint presentation,
- insert and delete slides in new presentation,
- view a presentation,
- using colors in presentation, and
- add header and footer

STRUCTURE:

7.1 Introduction

7.2 Getting Started

7.3 Basic Operation on Presentation
- **7.3.1 Creating New Presentation**
- **7.3.2 Save the Presentation**
- **7.3.3 Opening Existing Presentation**
- **7.3.4 Closing Presentation**

7.4 Inserting and Deleting Slides

7.5 Viewing a Presentation
- **7.5.1 Normal View**
- **7.5.2 Slide Sorter View**
- **7.5.3 Notes Page View**
- **7.5.4 Reading View**

7.6 Entering and Editing Text

7.7 Enhancing Text Presentation

7.8 Working with Color and Line Style

7.9 Adding Headers and Footers

7.10 Summary

7.11 Key Terms

7.12 Self-Assessment Questions

7.13 Suggestive Readings

7.1 INTRODUCTION:

Microsoft PowerPoint is a presentation software which is part of the Microsoft office software. PowerPoint is a commercial presentation application written and distributed by Microsoft.

PowerPoint makes it easy to create, collaborate, and present your ideas dynamically and visually. PowerPoint is used to create overhead transparency, 35mm slides, Photo Print or on-screen presentations. You can insert pictures, sounds, animation and type text in presentations along with integration with other Microsoft Office products like Excel.

There are Auto Layouts and templates that make the creation of presentation simple. PowerPoint offers a way to preview show, add special effects to the slides as displayed on-screen and rehearse the timings of each slide. By default, documents saved in PowerPoint 2010 are saved with the .pptx extension whereas the file extension of the prior PowerPoint versions is ppt

7.2 STARTING POWERPOINT:

In this section, you will understand how to get started with PowerPoint 2010. To access PowerPoint 2010, you must have Microsoft Office 2010 installed in your PC.

- To start **POWER POINT**,
- Click on the **Start** button
- Place the pointer on **Programs**; a sub-menu of Programs is displayed.
- In this menu, search for **Microsoft Office** and click on it.
- Now, select **Microsoft Office Power Point 2010** from the sub-menu and click it.

This will launch the Microsoft PowerPoint 2010 application and you will see the following presentation window

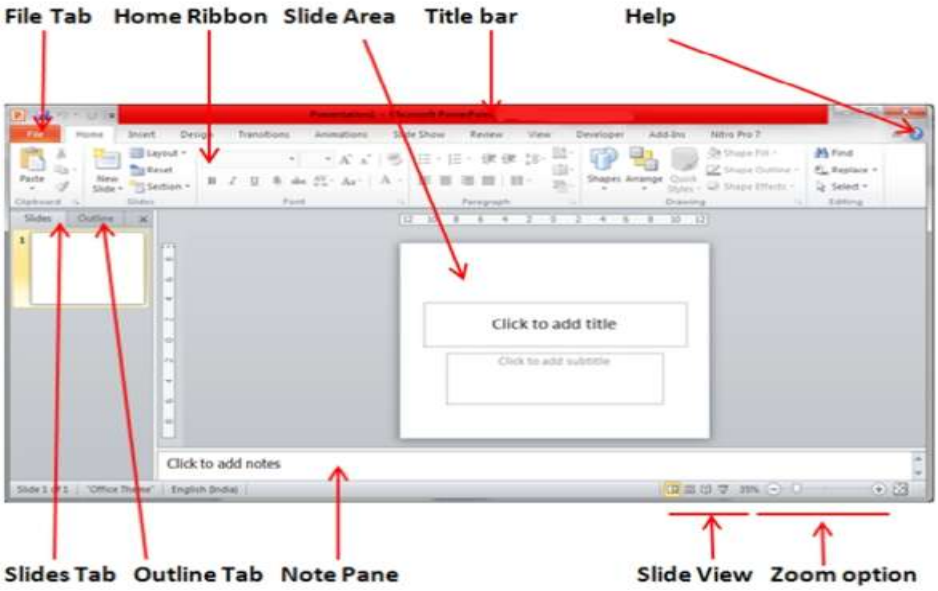


Figure 7.1 Overview of Slide

Screen Layout of Power Point 2010

You will see the default opening screen is actually composed of three parts:

- The Slides/Outline tabs on the left which displays thumbnails or lists of all slides
- The main Slide pane/area (center) which allows you to work on individual slides.
- Notes Pane (bottom): In the Notes pane, you can type notes about the current slide. You can distribute your notes to your audience or see your notes in Presenter view when you give presentation.

Here are some important features of standard PowerPoint Window which are necessary to learn before going in details of technical instructions on how to create a Power Point presentation. Let's get familiar with these features:

Title Bar

This is the top section of the window. It shows the name of the file followed by the name of the program which in this case is Microsoft PowerPoint.

Slide Area

This is the area where the actual slide is created and edited. You can add, edit and delete text, images, shapes and multimedia in this section.

Help

The Help Icon can be used to get PowerPoint related help anytime you need. Clicking on the "**?**" opens the PowerPoint Help window where you have a list of common topics to browse from. You can also search for specific topics from the search bar at the top

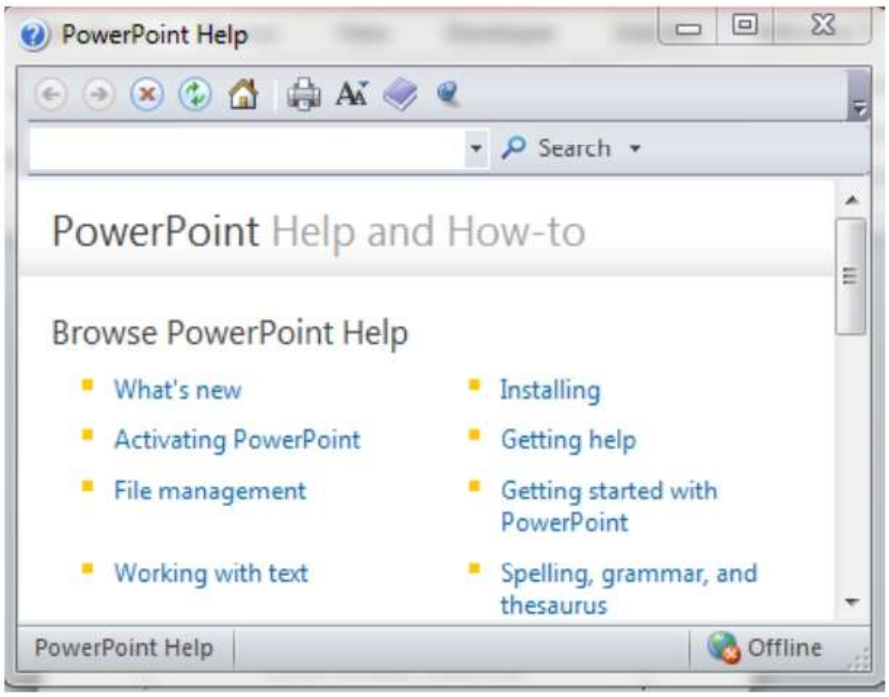


Figure 7.2 Help Window

Quick Access Toolbar

The Quick Access Toolbar is located just under the ribbon. This toolbar offers a convenient place to group the most commonly used commands in PowerPoint. It contains functions that you may want to use with ease. You can customize the Quick Access Toolbar in PowerPoint 2010 as similar to MS Word 2010

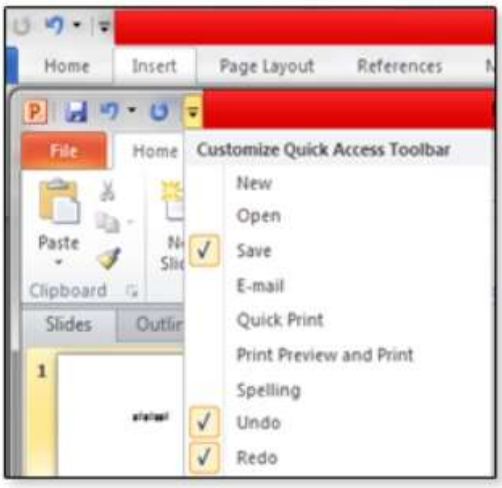


Figure 7.3 Quick Access Toolbar

Slides Tab
This section is available only in the Normal view. It displays all the slides in sequence. You can add, delete and reorder slides from this section. **Outline view**
It displays only the text of multiple slides in the outline form as shown in Fig.

4. A special set of tools appears to the left of the presentation text in the Outline view. A numbered slide icon is displayed to the left of each slide's title. If a slide contains no pictures or graphics, the slide icon is empty except for a narrow line near the top indicating the title. If the slide contains a picture or other objects, the slide icon also contains a graphical representation.

In this view, you can edit the displayed text, delete slides and even cut and paste slides from one location to another. Cutting and pasting is possible within the same presentation or between different presentations

Figure 7.4. Outline view

File Tab
You can use this tab for basic file operations such as opening, saving, sharing, exporting, printing and managing your presentation.You can create new presentations based on blank or predefined templates. It also consists ofoptions for settings in PowerPoint.

Ribbon

The ribbon provides all the features to you in MS PowerPoint. It contains main tabs and its subsequent features. Each tab ofRibbon is divided into Groups such as Home tab contains Clipboard, Slides, Font, Paragraph, Drawing and Editing. Each group has also a triangular icon in the lower right hand corner, clicking this will bring up an additional menu window. The Ribbon contains some other tab options such as Insert, Design, Transitions, Animations, Slide Show, Review and View

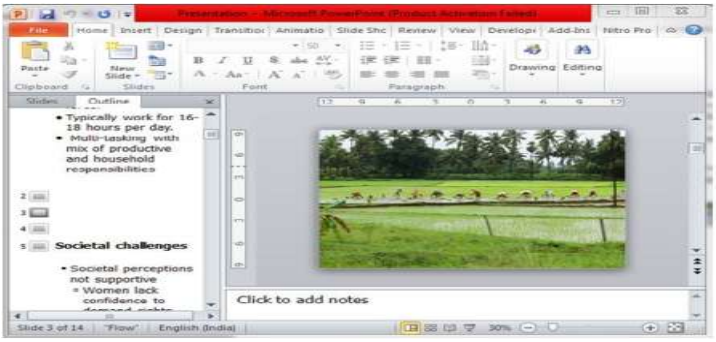


Figure 7.5 Home Tab

Home Tab

The Home tab includes many command groups such as Clipboard functions, manipulating slides, fonts, paragraph settings, drawing objects and editing functions. This group contains many commands for creating and manipulating presentation. For example, Font group is collection of drop-down menu and button options that are used for formatting a Presentation.

Insert

You can add variety of images, shapes, Chart, table, SmartArt and WordArt to the Presentation. The Insert tab is collection of commands related to tables, Illustrations, links, Header & Footer, text and Symbols.

Design

The Design tab has features that allow you to manipulate the Page Setup, background and Colour themes of a Presentation.

Animations

Using Animations tab, you can add, customise, and preview animations and transitions between slides.

Slide Show

The Slide Show tab is used for managing the setting for showing presentations. It allows you to select how your presentation will be displayed.

Review

The Review tab allows you to make changes to your documents and correct spelling and grammar errors

View
This tab features provides options to show the presentation. The view tab allows you to set the view of presentation (i.e. normal, slide sorter), show/ hide feature and zoom your presentation

7.3 BASIC OPERATION ON PRESENTATION:

In this section, you will learn about how to create a new presentation, open existing slides, saving and closing presentation.

7.3.1 Creating New Presentation

PowerPoint offers a variety of ways to create a new presentation. You can create the presentation by using a wizard or templates, or you can create blank presentations that contain no Colour or style or enhancements.

When you will open PowerPoint for the first time, the program should open a blank slide (refer to Fig. 9.1).

7.3.2 Creating a Blank Presentation

Blank Presentation creates the default templates called Default Design that uses no Colour and styles or enhancements. When you create a blank presentation, you have a full control of the Colour scheme, layout and style on your slides. You can add a template, Colours and other enhancements selectively at any time by using the menu or toolbar command

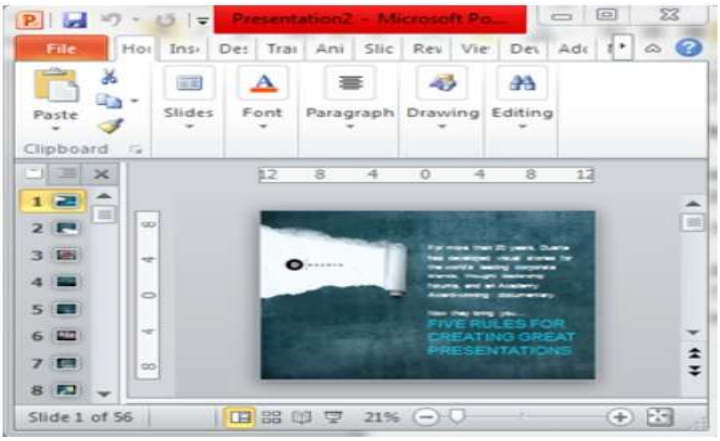


Figure 7.6 File Tab and their options

You can do the following for creating a new presentation:

- Select the **New** option from the **File tab**. Anew presentation dialog box is displayed as shown in Fig. 9.6.
- Select the **Blank presentation** and click on the **Create** button.
- A New presentation slide is displayed on the screen (see Fig. 9.1). **Creating a Presentation Using a Template**

When predefined slide and title, Color schemes and graphics elements are saved in a presentation file, it is called a presentation template. Using Installed themes option, you can use these templates to create new presentation with same formatting

You can do the following for creating a new presentation using a template:

- click on **File Tab** and select **new** command
- click on **Templates** (see Figure 9.6) or Installed Themes
- Select create button to display a new presentation.

Now, a new presentation is displayed on screen

Figure 7.7 New Presentation with Templates

Save the Presentation

You can save your presentation using Save or Save As option. The **Save** is used to save the presentation. For the first time, it will ask for filename and save as type. The **Save As** option may be used when you are sending the presentation to someone who have different version of PowerPoint or you want to store the presentation in a different location with different name

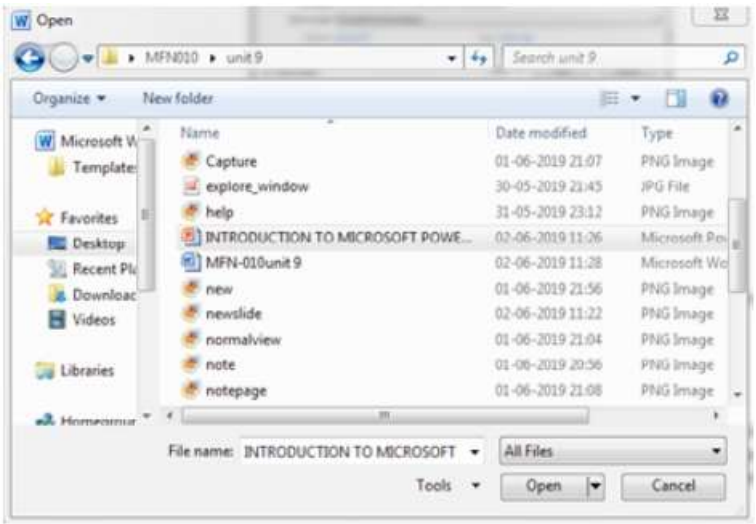


Figure 7.8 Save dialog box

To do this,

Click **File tab**

- Select **Save** or **Save As** and a dialog box will appear.
- Give filename with file type and select location of the filename
- Click on save button to store the presentation in hard disk. (see figure 9.8)

7.3.3 Opening Existing Presentation

You can open an existing power point presentation using **File** tab and **open** command. You can also use '**Recent Documents**' from File tab to open existing presentation which lists all current presentation.

When you select open command (see Fig. 9.6) from File tab to open existing presentation, a dialog box will appear (see Fig. 9.9), select the name of the presentation and click on open button to open a presentation.

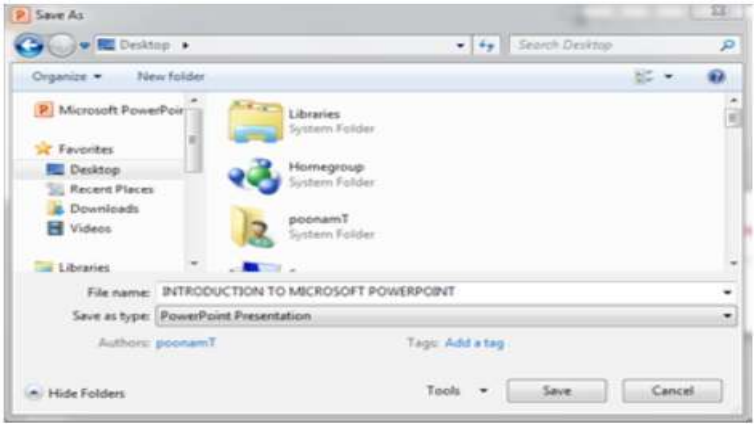


Figure 7.9 Open Dialog box for opening a Presentation

7.3.4 Closing Presentation

When you finished your work on a presentation, you can close it. Closing a presentation means it frees up computer memory for other activities. For closing presentation, click the **File tab** and **Close**

Figure 7.10 Close dialog box

If the current presentation is unsaved, PowerPoint will display a dialog box, as shown in Fig. 9.10 asking to save the current presentation with Yes and No options. Select 'Yes', if you

want to save file, No if you do not want to save the file or cancel to return to your file without saving it.

7.4 INSERTING AND DELETING SLIDES:

In previous section, you have created presentation. After this, you can insert additional slide or delete slides wherever necessary in your presentation.

Inserting Additional Slides

Using the following, you can insert additional new slide to the presentation. For this, do the following:

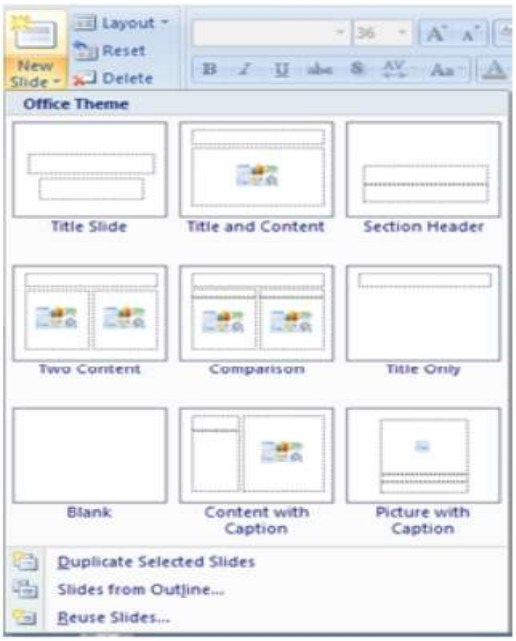


Figure 7.11 New Slide Options

- Select **New Slide** option on Slides group of **Home tab.**
- After this operation, you will find more new slides format. You can select any of them according to your need such as title slide, title with content, Section header, blank slide or duplicate slide. (see **Fig. 9.11**)

Deleting Slides

To delete the slide, do the following:

- Select the slide to be deleted ‘and right click on it. (see **Fig. 9.12**)
- A shortcut menu will be open and select Delete Slide option.
- Your selected slide will be deleted from presentation.

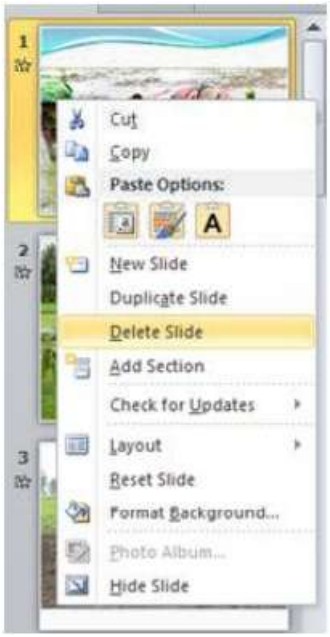


Figure 7.12 Delete Slide

7.5 VIEWING A PRESENTATION:

There are many views to view a PowerPoint Presentation. Each view has a particular purpose and advantage. You can either select the Presentation Views option from the View tab on Ribbon or select views using the view buttons located to the left of the Zoom control, near the bottom of the screen

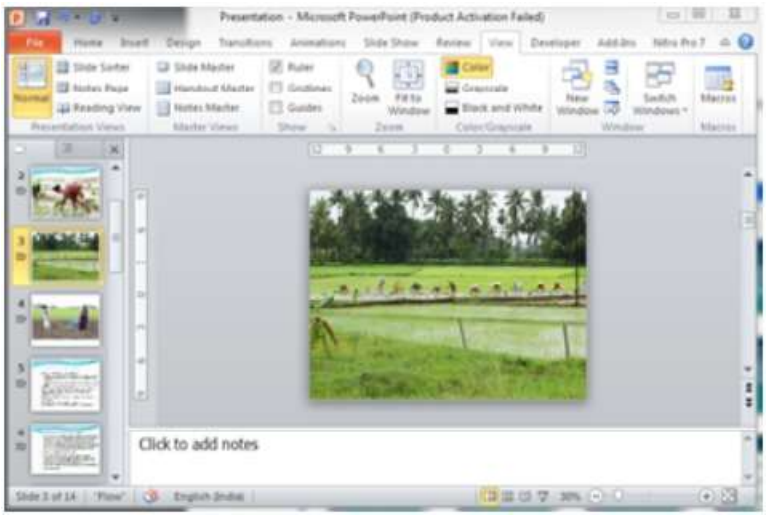


Figure 7.13 Viewing Presentation options

The four different Presentation Views (see Fig. 9.13) available in PowerPoint are as follows:

- Normal view
- Slide Sorter View
- Notes Page

- Reading view

7.5.1 Normal view

This displays page in normal view with the slide on the right and a list of thumbnails to the left.

Figure 7.14 Normal View

This view allows you to edit in dividual slides and also rearrange them. It displays all the slides in sequence. This view also has Notes Page Pane at the bottom of the screen. This is the default view when you first open PowerPoint. In this view, all the menus and toolbars are available for use. You can **add**, **delete** and **reorder** slides from this section

7.5.2 Slide Sorter view

Slide Sorter view gives you a miniature picture of each slide. This view only allows you to rearrange the slides but not edit the contents of each slide

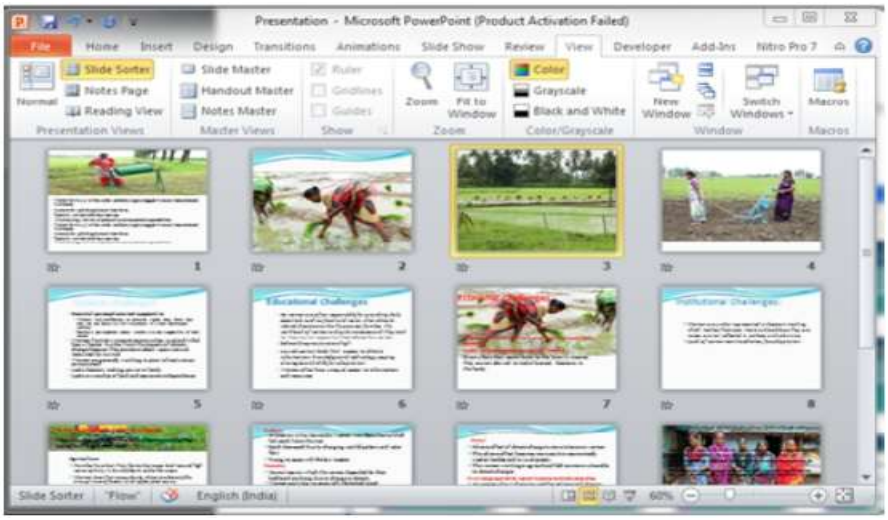


Figure 7.15 Slide Sorter view

7.5.3 Notes Page

This section allows you to add notes for the presentation. These notes will not be displayed on the screen during the presentation; these are just quick reference for the presenter.

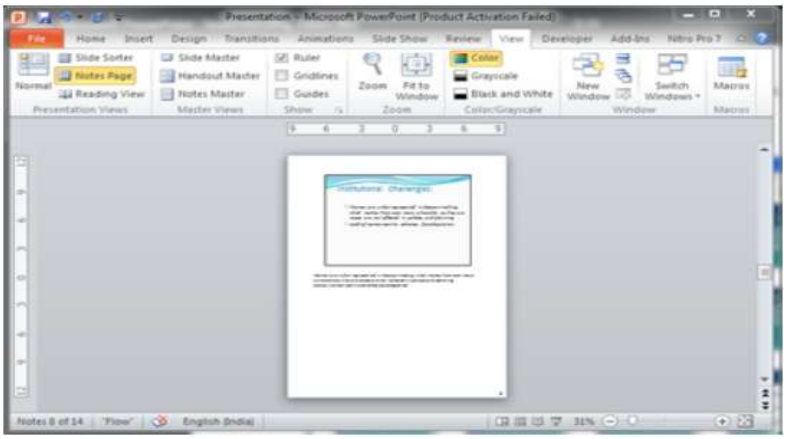


Figure 7.16 Note Page View

7.5.4 Reading View

Reading View shows the slide that fits within the window and you can change to this view when you want to proof read or rehearse

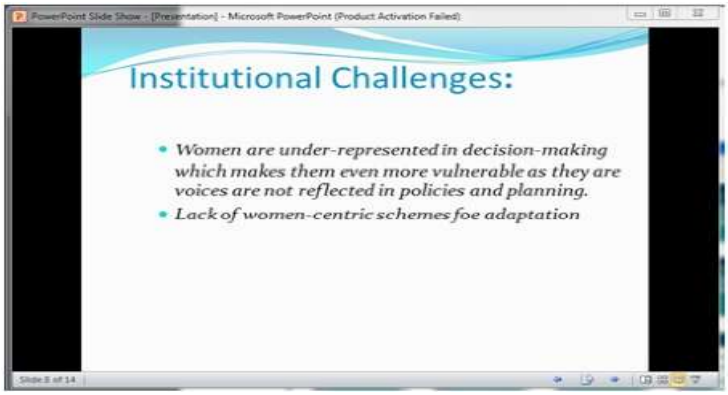


Figure 7.17: Reading View

Zoom Options

PowerPoint displays the Presentation at a present percentage of its full size. The zoom control lets you zoom in for a closer look at your text. The zoom control consists of a slider that you can slide left or right to zoom in or out, you can click on the - and + buttons to increase or decrease the zoom factor. The maximum zoom supported by PowerPoint is 400% and the 100% is indicated by the mark in the middle.

The percentage PowerPoint uses varies, depending on your video driver, the screen resolution you use, and the size of your monitor.

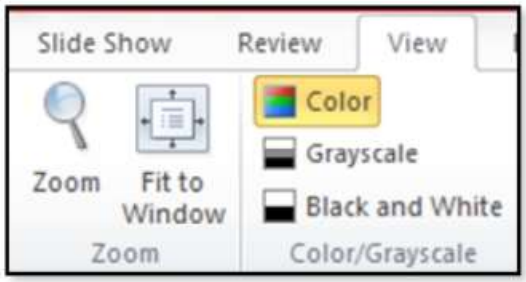


Figure 7.18 Zoom option on View tab

To specify the zoom percentage:

- Select the Zoom option from the Zoomgroup under View tab on Ribbon.

- A zoom dialog box is displayed as shown in **Fig. 9.19**

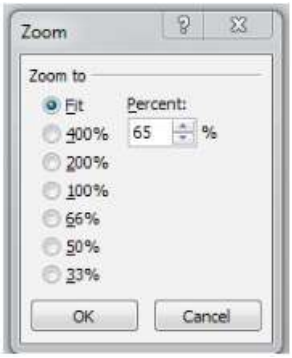


Figure 7.19 Zoom window

- Select the percentage.
- Click on the OK button

7.6 ENTERING AND EDITING TEXT:

Text is one of the most important components of any application. Every slide in a Presentation contains text of some kind, even if it is just the title. Entering and editing text in PowerPoint is similar to entering and editing text in any Office application.

Entering text using Title content layout of slide is simples. You must use it as far as possible. In this case, when you insert a new slide, select its layout as title at content and then click on title to enter the title and in the content to enter the bulleted list of points. If you want to add new text objects other than the bulleted list of points. If you want to add new text objects other than bulleted points, you can follow the following process:

- Select the Text Box button from the Text group of Insert tab on Ribbon.
- Place the pointer where you want the text box to appear.
- Click on the left mouse button. A text box is displayed with the insertion point as show in Fig. 9.20.
- Type the desired text in the text box. The text box expands to accommodate the text you enter. If you want to type on a new line, press Enter.

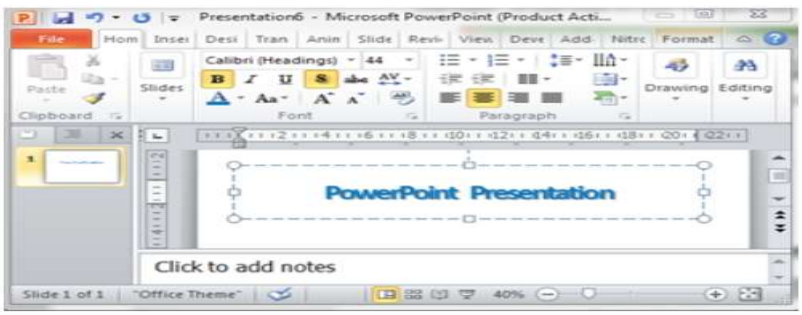


Figure 7.20 New Text Object creation

Changing Text and Correcting Errors

Click on the text object to make changes. An insertion point is displayed, indicating that you can edit the text. You can use the Cut, Copy, Paste options from the Clipboard group of Home tab..

When you finish editing text, deselect the object by clicking on a blank area of the slide or the gray border around the slide

7.7 ENHANCING TEXT PRESENTATION:

Enhancing Text

PowerPoint Provides many text enhancing functions including changing placements, selecting fonts, changing text style, color, size and adding special effects like underline, shadow, and so on.

Changing Line and Paragraph Spacing
PowerPoint helps you to set the spacing between lines as well as the space before and after paragraphs.

- Select the text for which you want to change line or paragraph spacing, either in the master or on the individual slide.
- Select the **Home** tab.
- Under this tab, click on right corner of the **Paragraph group** option.
- A window will appear in which you can give appropriate value for spacing between lines as well as the space before and after paragraphs **(see Fig. 9.21)**

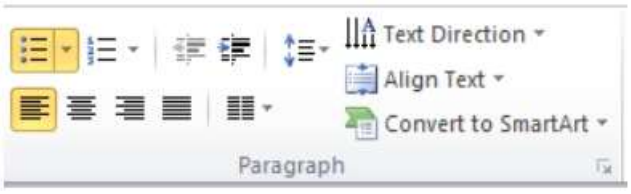


Figure 7.21 Paragraph Spacing option

Aligning Text

Alignment is the way text is placed between the margin of a slide or a text box. In presentation slides, text is generally left-aligned for paragraphs or bullets. You can align text using the buttons given below on the Paragraph Group option under Home tab. You can align your text as left, center, right and Justify

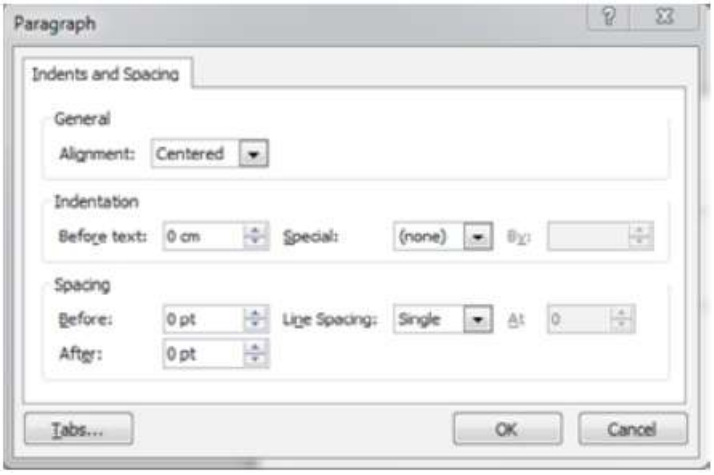


Figure 7.22 Alignment option

7.8 WORKING WITHCOLOUR ANDLINESTYLE:

All objects that you draw in PowerPoint except a line have a fill Colour, line Colour and line style. For any object, you can turn off the line colour and fill colour. In most templates, an object's line is a narrow solid line. You can select any one of the wider line styles or any one of the double or triple lines. You can change a solid line to dashed, dotted, and soon, by selecting one of the dash style options. You can also add arrow heads to either end or to both ends of the line or arc

Choosing Fill Color, Line Colors and Line Styles

You can change an object's Fill Color in PowerPoint.

- Select the object.
- After your selection, a Format tab will enable as drawing toolbar.
- Click on it. It will show many group options such as Insert Shapes, Shape style, WordArt Styles.
- Here, you can select any command like Shape Fill, Shape Outline, Shape effects, Text Fill, Text outline and Text Effects

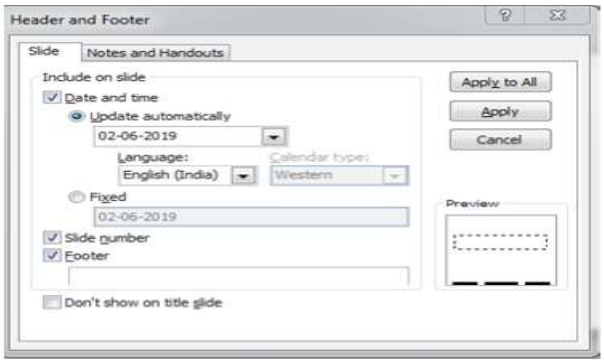


Figure 7.23 Format tab

7.9 ADDING HEADERS AND FOOTERS:

Header is a text element shown at the top margin of a document while footer is a text element shown at the bottom margin. You can add header and footer to your slides, presentation notes and audience handouts. Headers and footers appear on every slide. You can choose not to have them appear on the title slide. They often include information such as the presentation title, slide number, date, and name of the presenter.

You can do the following for adding Header and Footer in your presentation:

- Click the Insert tab, and then click the Header & Footer button under the Text group.
- A window will open (**see Fig. 9.24**). Here, you can click the Slide or Notes and Handouts tab.
- Enter or select the information you want to include on your slide or your notes and handouts.
- If you do not want to include a header and footer on the title slide, select the 'Don't show on title slide' check box.
- Click ''Apply" to apply your selections to the current slide (if available), or click ''Apply to All" to apply the selections to all slides

Figure 7.24 Header and Footer

7.10 SUMMARY:

PowerPoint presentation enhancements and slideshow features constitute a vital framework for effective visual communication in educational, corporate, and professional settings. This structured approach focuses on transforming basic slides into dynamic, engaging experiences through strategic implementation of advanced PowerPoint capabilities.

7.11 KEY TERMS:

Normal View, Slide Sorter View, Reading View, Slide Show View, Notes Page View, Presenter View

7.12 SELF-ASSESSMENT QUESTIONS:

1. What view in PowerPoint allows you to see your speaker notes while presenting, but keeps them hidden from your audience?
2. Which view displays thumbnails of all slides to help with reorganization and provides the easiest way to rearrange slide order?
3. What is the default view in PowerPoint that shows your current slide, thumbnails, and notes panel?
4. How do you access Slide Master view, and what is its primary purpose?
5. What key do you press to start a presentation from the beginning?
6. What's the difference between Reading View and Slide Show View?

7.13 SUGGESTIVE READINGS:

1. "Presentation Zen" by Garr Reynolds - Focuses on design principles and effective visual communication for presentations
2. "Slide" by Nancy Duarte - A comprehensive guide to creating effective visual presentations
3. "PowerPoint 2019 For Dummies" by Doug Lowe - Practical guide covering all PowerPoint fundamentals including different views
4. "Better PowerPoint" by Stephen Kosslyn - Evidence-based approach to creating effective slides

Onne Resources

Microsoft's Official PowerPoint Support - Tutorials and documentation on all PowerPoint features
https://support.microsoft.com/en-us/powerpoint
LinkedIn Learning (formerly Lynda.com) - Multiple courses on PowerPoint from beginner to advanced levels

Dr D.V. Chandrashekar

LESSON- 8

MS EXCEL: WORKSHEET CREATION, FORMATTING, AND FUNCTIONS

OBJECTIVES:

1. Build functional spreadsheets for practical business applications
2. Troubleshoot and debug formula errors
3. Document worksheets effectively for future reference and collaboration
4. Implement data validation to ensure data integrity
5. Create dynamic models that update automatically when input data changes

STRUCTURE:

8.1 Introduction

8.2 Entering Excel Formulas and Formatting Data
- **8.2.1 Perform Mathematical Calculations**
  - **8.2.1.1 Addition, Subtraction, Multiplication and Division of Numbers**
  - **8.2.2.2 Perform Advanced Mathematical Calculations**
  - **8.2.2.3 AutoSum**
  - **8.2.2.4 Align Cell Entries**
  - **8.2.2.5 Copy, Cut and Paste**

8.3 Creating Excel Functions, Filling Cells, and Printing
- **8.3.1 Understanding Functions**
- **8.3.2 Functions**
- **8.3.3 Calculate an Average**
- **8.3.4 Find the Lowest Number**
- **8.3.5 Find the Highest Number**
- **8.3.6 Fill Cells Automatically**
- **8.3.7 Set Print Options**
- **8.3.8 Advanced Functions**

8.4 Summary

8.5 Key Terms

8.6 Self-Assessment Questions

8.7 Suggestive Readings

8.1 INTRODUCTION TO MS EXCEL:

Microsoft Excel is an electronic spreadsheet. You can use it to organize your data into rows and columns. You can also use it to perform mathematical calculations quickly. This course teaches Microsoft Excel basics as a prelude to the use of Statistical Analysis System (SAS) software in carrying out more complex statistical analysis. Although knowledge of how to navigate in a Windows environment is helpful, this manual is created for the computer novice.

At the end of the course, participants are expected to know how to use Microsoft Excel to:

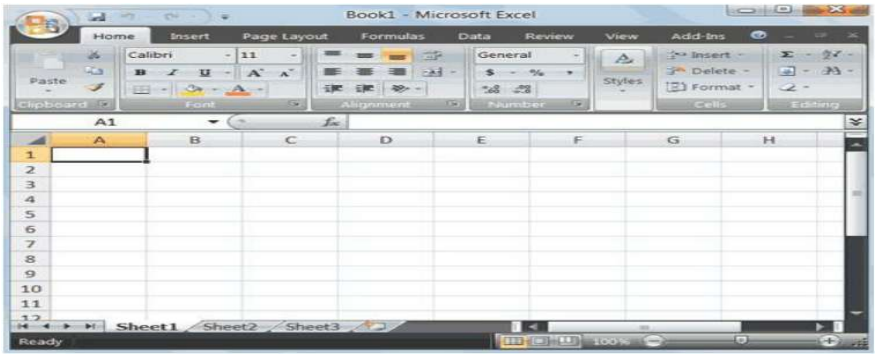
- Enter text and numbers in a spreadsheet
- Enter Excel formulas
- Format data
- Create Excel functions
- Fill cells automatically
- Print results
- Enter advanced Excel formulas

Entering Text and Numbers

The Microsoft Excel Window

This Section will introduce you to the Excel window. To begin this Section, start Microsoft Excel 2007 as follows:

1. Click on Microsoft Start Button
2. Point the mouse on All Programs
3. Click on Microsoft Office
4. Click on Microsoft Excel 2007



In the upper-left corner of the Excel 2007 window is the Microsoft Office button. When you click the button, a menu appears. You can use the menu to create a new file, open an existing file, save a file, print and perform many other tasks

The Quick Access Toolbar

Next to the Microsoft Office button is the Quick Access toolbar. The Quick Access toolbar gives you quick access to commands you frequently use .

Next to the Quick Access toolbar is the Title bar. On the Title bar, Microsoft Excel displays the name of the workbook you are currently using. At the top of the Excel window, you should see "Book 1 - Microsoft Excel" or a similar name.

The Ribbon

In Microsoft Excel 2007, you use the Ribbon to issue commands. The Ribbon is located near the top of the Excel window, below the Quick Access toolbar

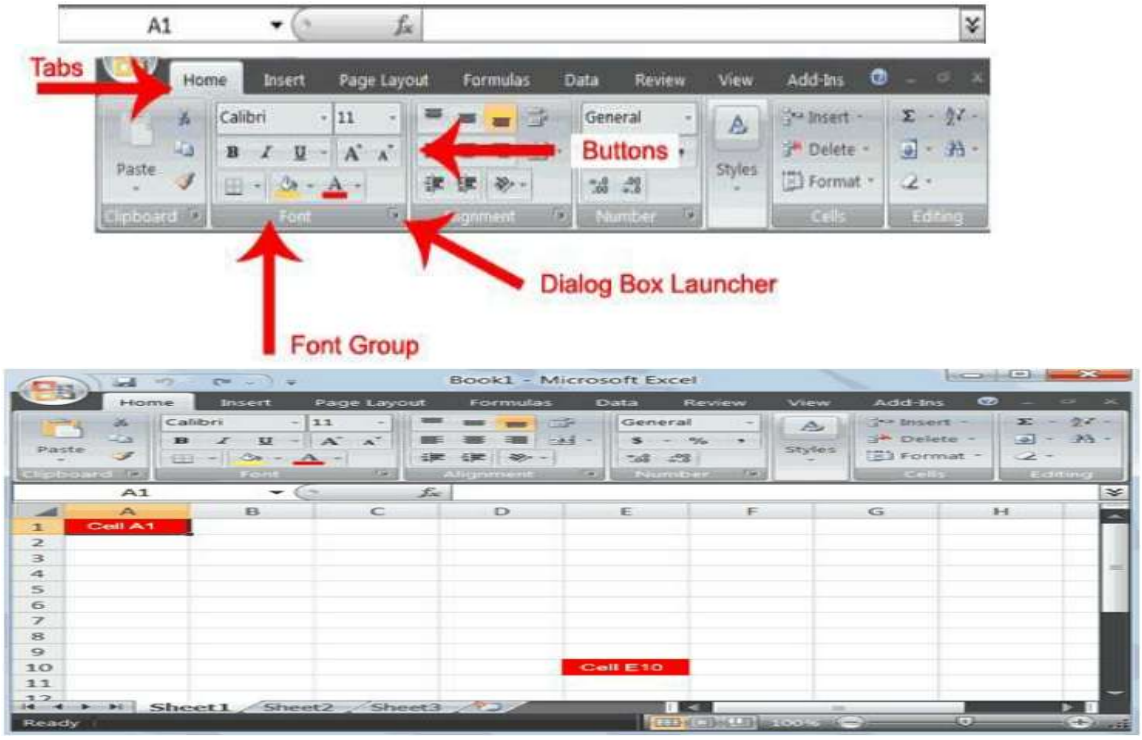


Figure 8.2 Worksheet Menu

Microsoft Excel consists of worksheets. Each worksheet contains columns and rows. The columns are lettered A to Z and then continuing with AA, AB, AC and so on; the rows are numbered 1 to 1,048,576.

The combination of a column coordinate and a row coordinate make up a cell address. For example, the cell located in the upper-left corner of the worksheet is cell A1, meaning column A, row 1. Cell E10 is located under column E on row 10. You enter your data into the cells on the worksheet

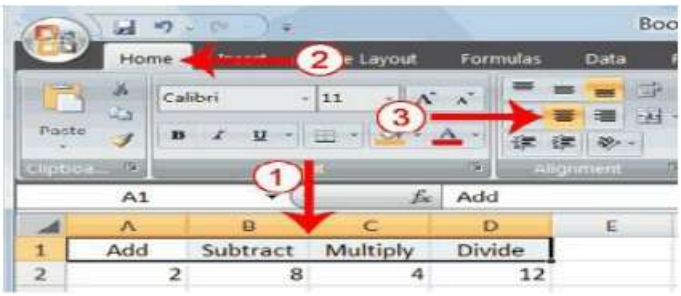


Figure 8.3 The Formula Bar

If the Formula bar is turned on, the cell address of the cell you are in displays in the Name box which is located on the left side of the Formula bar. Cell entries display on the right side of the Formula bar.

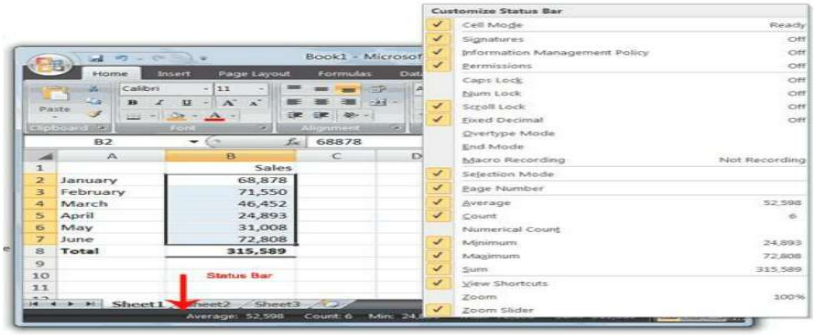


Figure 8.4 The Status Bar

The Status bar appears at the very bottom of the Excel window and provides such information as the sum average, minimum, and maximum value of selected numbers.

Save a File

This is the end of Section 1. To save your file:

1. Click the Office button. A menu appears.
2. Click Save. The Save As dialog box appears.
3. Go to the directory in which you want to save your file.
4. Type **Section1** in the File Name field.
5. Click Save. Excel saves your file

Close Excel Close Microsoft Excel.

1. Click the Office button. A menu appears.
2. Click Close. Excel closes

8.2: ENTERING EXCEL FORMULAS AND FORMATTING DATA:

Section 1 familiarized you with the Excel 2007 window, taught you how to move around the window, and how to enter data. A major strength of Excel is that you can perform mathematical calculations and format your data. In this Section, you will learn how to perform basic mathematical calculations and how to format text and numerical data. To start this Section, open Excel.

8.2.1 Perform Mathematical Calculations

In Microsoft Excel, you can enter numbers and mathematical formulas into cells. Whether you enter a number or a formula, you can reference the cell when you perform mathematical calculations such as addition, subtraction, multiplication, or division. When entering a mathematical formula, precede the formula with an equal (=) sign. Use the following to indicate the type of calculation you wish to perform:

+	Addition
–	Subtraction
*	Multiplication
/	Division
^	Exponential

In the following exercises, you practice some of the methods you can use to perform mathematical calculations.

8.2.1.1 Addition, Subtraction, Multiplication and Division of Numbers

1. Type: Add, Subtract, Multiply, and Divide in cells A1, B1, C1, and D1 respectively
2. Type: 12, 25, 11 and 75 in cells A2, B2, C2 and D2 respectively
3. Type: 8, 13, 6 and 5 in cells A3, B3, C3 and D3 respectively
4. Type: = A2 + A3 in cell A5 and press Enter
5. Type: = B2 + B3 in cell A5 and press Enter
6. Type: = C2 + C3 in cell A5 and press Enter
7. Type: = D2 + D3 in cell A5 and press Enter

8.2.2.2 Perform Advanced Mathematical Calculations

When you perform mathematical calculations in Excel, be careful of precedence. Calculations are performed from left to right, with multiplication and division performed before addition and subtraction.

Advanced Calculations

1. Move to cell A7.
2. Type =3+3+12/2*4.
3. Press Enter.

Note: Microsoft Excel divides 12 by 2, multiplies the answer by 4, adds 3, and then adds another 3. The answer, 30, displays in cell A7

8.2.2.3 AutoSum

You can use the AutoSum button on the Home tab to automatically add a the data

AutoSum

The following illustrates AutoSum

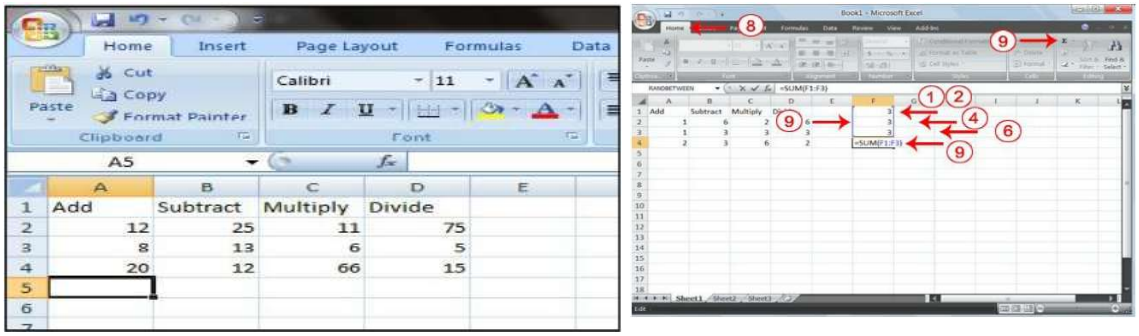


Figure 8.5 Example of Arthematic operations

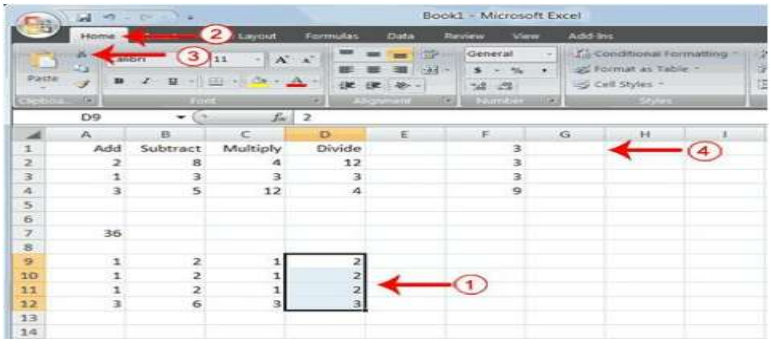


Figure 8.6 Auto Sum in Menu Bar

1. Go to cell F1.
2. Type 3.
3. Press Enter. Excel moves down one cell.
4. Type 3.
5. Press Enter. Excel moves down one cell.
6. Type 3.
7. Press Enter. Excel moves down one cell to cell F4.
8. Choose the Home tab.
9. Click the AutoSum button in the Editing group. Excel selects cells F1 through F3 and enters a formula in cell F4.

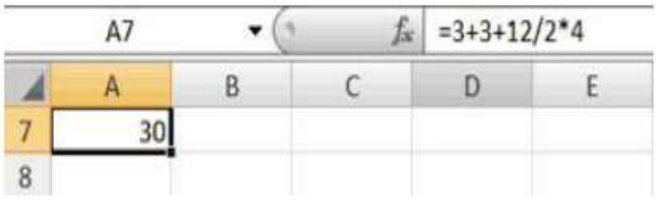


Figure 8.7 Arthematic Operations in formula Bar

Note that you can click on the arrow next to AutoSum to access other automatic calculations like average, minimum and maximum values, count numbers, etc

8.2.2.4 Align Cell Entries

When you type text into a cell, by default your entry aligns with the left side of the cell. When you type numbers into a cell, by default your entry aligns with the right side of the cell. You can change the cell alignment. You can center, left-align, or right-align any cell entry. Look at cells A1 to D1. Note that they are aligned with the left side of the cell

To centre cells A1 to D1

1. Select cells A1 to D1.
2. Choose the Home tab
3. Click the Centre button in the Alignment group. Excel centres each cell's content

8.2.2.5 Copy, Cut and Paste

1. Select cells D9 to D12
2. Choose the Home tab
3. Click the Cut button
4. Move to cell G1
5. **Click the Paste button** Excel moves the contents of cells D9 to D12 to cells G1 to G4

8.3 CREATING EXCEL FUNCTIONS, FILLING CELLS, AND PRINTING:

By using functions, you can quickly and easily make many useful calculations, such as finding an average, the highest number, the lowest number, and a count of the number of items in a list. Microsoft Excel has many functions that you can use.

Using Reference Operators

To use functions, you need to understand reference operators. Reference operators refer to a cell or a group of cells. There are two types of reference operators: range and union.

A range reference refers to all the cells between and including the reference. A range reference consists of two cell addresses separated by a colon. The reference A1:A3 includes cells A1, A2, and A3. The reference A1:C3 includes cells A1, A2, A3, B1, B2, B3, C1, C2, and C3**.**

A union reference includes two or more references. A union reference consists of two or more numbers, range references, or cell addresses separated by a comma. The reference A7, B8:B10, C9,10 refers to cells A7, B8 to B10, C9 and the number 10.

8.3.1 Understanding Functions

Functions are prewritten formulas. Functions differ from regular formulas in that you supply the value but not the operators, such as +, -, *, or /. For example, you can use the SUM function to add. When using a function, remember the following:

1. Use an equal sign to begin a formula.
2. Specify the function name.
3. Enclose arguments within parentheses. Arguments are values on which you want to perform the calculation. For example, arguments specify the numbers or cells you want to add.
4. Use a comma to separate arguments. Here is an example of a function:
   =SUM (2,13, A1, B2:C7)

In this function, known as the SUM function:
1. The equal sign begins the function.
2. SUM is the name of the function.
3. 2, 13, A1, and B2:C7 are the arguments. Parentheses enclose the arguments.
4. Commas separate the arguments.

After you type the first letter of a function name, the AutoComplete list appears. You can double-click on an item in the AutoComplete list to complete your entry quickly. Excel will complete the function name and enter the first parenthesis

8.3.2 Functions

The SUM function adds argument values

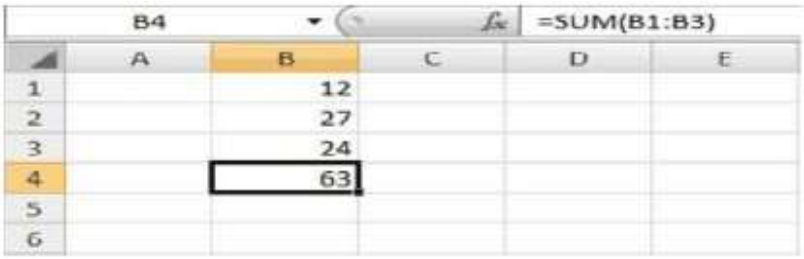


Figure 8.8 Sum function

1. Open Microsoft Excel.
2. Type 12 in cell B1.
3. Press Enter.
4. Type 27 in cell B2.
5. Press Enter.
6. Type 24 in cell B3.
7. Press Enter.

8. Type =SUM(B1:B3) in cell A4.
9. Press Enter. The sum of cells B1 to B3, which is 63, appears

8.3.3 Calculate an Average

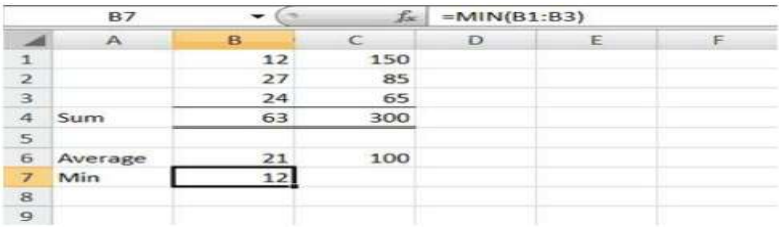


Figure 8.9 Average function

1. Move to cell A6.
2. Type Average. Press the right arrow key to move to cell B6.
3. Type =AVERAGE (B1:B3).
4. Press Enter. The average of cells B1 to B3, which is 21, appears

8.3.4 Find the Lowest Number

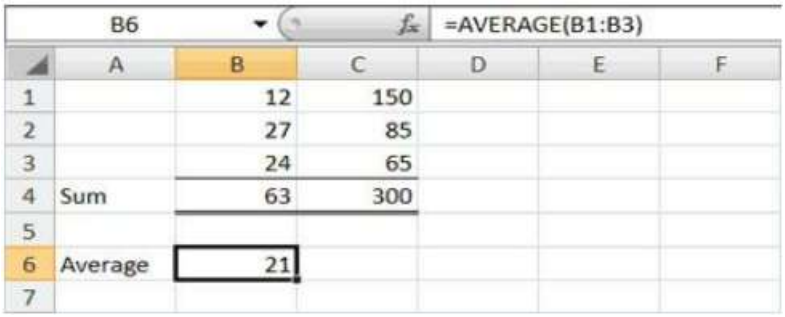


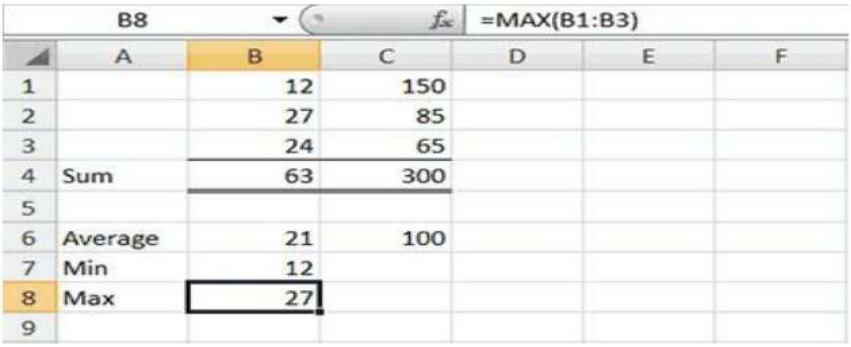
Figure 8.10 Min function

1. Move to cell A7.
2. Type Min. Press the right arrow key to move to cell B7.
3. Type =MIN (B1:B3).
4. Press Enter. The lowest number in the series, which is 12, appears

8.3.5 Find the Highest Number

You can use the MAX function to find the highest number in a series of numbers

1. Move to cell A8.
2. Type Max. Press the right arrow key to move to cell B8.
3. Type =MAX (B1:B3).
4. Press Enter. The highest number in the series, which is 27, appears

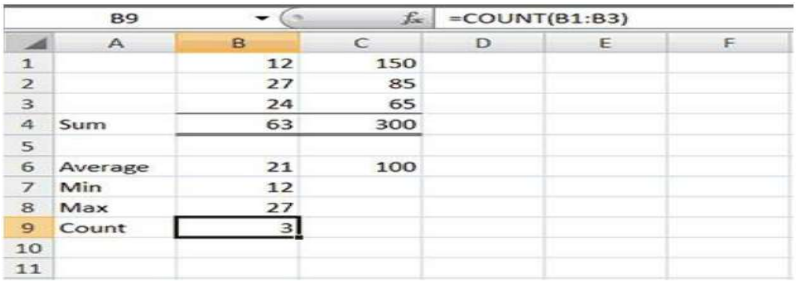


	A	B	C	D	E	F
1		12	150			
2		27	85			
3		24	65			
4	Sum	63	300			
5						
6	Average	21	100			
7	Min	12				
8	Max	27				
9						

Figure 8.11 MAX function

Count the Numbers in a Series of Numbers

1. Move to cell A9.
2. Type Count. Press the right arrow key to move to cell B9.
3. Type =COUNT (B1:B3).
4. Press Enter. The number of items in the series, which is 3, appears



	A	B	C	D	E	F
1		12	150			
2		27	85			
3		24	65			
4	Sum	63	300			
5						
6	Average	21	100			
7	Min	12				
8	Max	27				
9	Count	3				
10						
11						

Figure 8.12 Count function

8.3.6 Fill Cells Automatically

The following demonstrates filling the days of the week

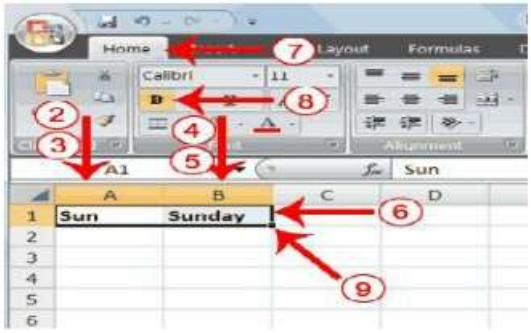


Figure 8.13 Fill Cells Function

(a) Fill Cell Function

1. Click the Sheet2 tab. Excel moves to Sheet2.
2. Move to cell A1.
3. Type Sun.
4. Move to cell B1.
5. Type Sunday.
6. Select cells A1 to B1.
7. Choose the Home tab.
8. Click the Bold button . Excel bolds cells A1 to B1.

9. Find the small black square in the lower-right corner of the selected area. The small black square is called the fill handle.
10.Grab the fill handle and drag with your mouse to fill cells A1 to B14. Note how the days of the week fill the cells in a series. Also, note that the Auto Fill Options button appears

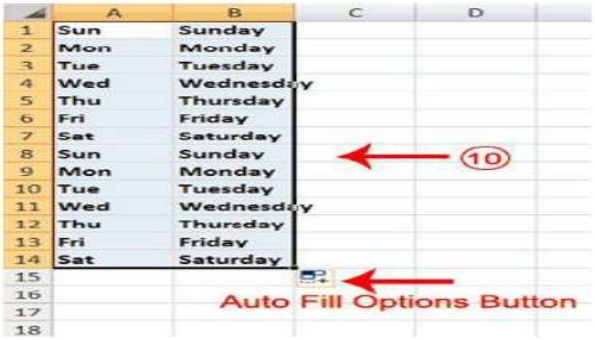


Figure 8. 14 Fill Function

(b) Fill Times

The following demonstrates filling time:

1. Type 1:00 into cell C1.
2. Grab the fill handle and drag with your mouse to highlight cells C1 to C14. Note that each cell fills, using military time.
3. Press Esc and then click anywhere on the worksheet to remove the highlighting.

To change the format of the time:

1. Select cells C1 to C14.
2. Choose the Home tab**.**
3. Click the down arrow next to the number format box a menu appears.
4. Click Time. Excel changes the format of the time**.**

(c) Fill Numbers

You can also fill numbers.
1. Type a 1 in cell D1.
2. Type a 2 in cell D2.
3. Select cells D1:D2
4. Grab the fill handle and drag with your mouse to highlight cells D1 to D14.
5. The cells fill as a series, starting with 1, 2, 3.

Here is another interesting fill feature.
1. Go to cell E1.
2. Type Section 1.
3. Grab the fill handle and drag with your mouse to highlight cells E1 to E14. The cells fill in as a series: Section 1, Section 2, Section 3, and so on

8.3.7 Set Print Options

There are many print options. You set print options on the Page Layout tab. Among other things, you can set your margins, set your page orientation, and select your paper size.

Margins define the amount of white space that appears on the top, bottom, left, and right edges of your document. The Margin option on the Page Layout tab provides several standard margin sizes from which you can choose.

Paper comes in a variety of sizes. Most business correspondence uses 8 ½ by 11 paper, which is the default page size in Excel. If you are not using 8 ½ by 11 paper, you can use the Size option on the Page Layout tab to change the Size setting

8.3.8 Advanced Functions

The SUMIF Function

Syntax

SUMIF(range,criteria,sum_range)

Range is the range of cells where Excel searches for the criteria that you want evaluated. Cells in each range must be numbers or names, arrays, or references that contain numbers. Blank and text values are ignored.

Criteria is the criteria in the form of a number, expression, or text that defines which cells will be added. For example, criteria can be expressed as 32, "32", ">32", or "apples".

Sum range are the actual cells to add if their corresponding cells in range match criteria. If sum range is omitted, the cells in range are both evaluated by criteria and added if they match criteria.

Note: The SUMIF function can be read as:
"Sum or add up sum range if range meets criteria**."**
=SUMIF (A2:A5,">160000", B2:B5)

The AVERAGEIF Function

Returns the average (arithmetic mean) of all the cells in a range that meet a given criteria.
Syntax
AVERAGEIF (range, criteria, average range)
Range is one or more cells to average, including numbers or names, arrays, or references that contain numbers.
Criteria is the criteria in the form of a number, expression, cell reference, or text that defines which cells are averaged. For example, criteria can be expressed as 32, "32", ">32", "apples", or B4.
Average range is the actual set of cells to average. If omitted, range is used.

8.4 SUMMARY:

- Worksheet **Creation** provides the foundation for organizing data in a structured manner, using rows, columns, and cells to create a framework for analysis
- Formatting turns raw data into readable, professional-looking spreadsheets by applying styles, colors, borders, and conditional formatting that highlight important information

- Functions are the analytical engine of Excel, allowing users to perform calculations ranging from simple arithmetic to complex statistical analysis, financial modeling, and data lookup operations

8.5 KEY TERMS:

Workbook, Worksheet/Sheet, Cell, Cell Reference, Range, Formula Bar, Name Box

8.6 SELF-ASSESSMENT QUESTIONS:

1. What is the difference between a workbook and a worksheet in Excel?
2. How do you insert a new worksheet between two existing worksheets?
3. Explain the difference between relative and absolute cell references. When would you use each?
4. What are the steps to rename a worksheet?
5. How do you freeze panes in Excel, and why might this be useful?
6. Name three types of number formats available in Excel and explain when each would be appropriate.
7. How do you apply conditional formatting to highlight cells that contain values greater than 100?
8. What is the purpose of cell styles in Excel?
9. Describe how to create a custom number format.
10. How do you merge cells, and what are potential drawbacks of doing so?
11. Write a formula that would calculate the sum of values in cells A1 through A10.
12. Explain the difference between COUNT, COUNTA, and COUNTIF functions.
13. Create an IF function that displays "Pass" if a student's score in cell B2 is greater than or equal to 60, and "Fail" if it's below 60.
14. How would you use VLOOKUP to find information in a table? Give an example.
15. Write a formula that calculates the average of a range of cells while ignoring zero values.
16. What is the purpose of the INDEX and MATCH functions, and how do they differ from VLOOKUP?
17. How would you use nested IF functions to assign letter grades (A, B, C, D, F) based on numerical scores?
18. Explain how to use data validation to create a dropdown list in a cell.
19. What is the purpose of the $ symbol in cell references (e.g., A1 vs. A1)?
20. How would you use Excel's SUMIF function to sum values that meet specific criteria?
21. Describe the steps to create a basic monthly budget worksheet from scratch.
22. How would you format a large dataset to make it more readable and highlight important information?
23. Create a formula that calculates the percentage change between two values.
24. How would you set up a worksheet to track inventory, including formulas for reorder alerts?
25. Explain how you would use Excel to analyse sales data and create a summary report.

8.7 SUGGESTIVE READINGS:

Books

1. **"Excel 2019 Bible" by Michael Alexander, Richard Kusleika, and John Walkenbach**
   Comprehensive guide covering all Excel fundamentals and advanced features

2. **"Microsoft Excel 2019 Data Analysis and Business Modeling" by Wayne Winston**
   Focuses on practical applications of Excel functions for business analysis
3. **"Excel Formulas & Functions For Dummies" by Ken Bluttman**
   Accessible reference for mastering Excel's calculation capabilities
4. **"Ctrl+Shift+Enter: Mastering Excel Array Formulas" by Mike Girvin**
   Deep dive into advanced Excel formulas and techniques
5. **"M is for (Data) Monkey" by Ken Puls and Miguel Escobar**
   Guide to Excel's Power Query for data transformation

Online Courses

1. **LinkedIn Learning's "Excel Essential Training" series**
   Structured learning path from basics to advanced features
2. **Coursera's "Excel Skills for Business" Specialization by Macquarie University**
   Four-part series covering essential through advanced Excel skills

Dr D.V. Chandrashekar

LESSON-9

MS-EXCEL

OBJECTIVES:

1. Understand the fundamental concepts and techniques of data sorting in Microsoft Excel, including single-column and multiple-column sorting.
2. Explore the process of filtering data in Excel using single and multiple criteria to refine datasets effectively.
3. Learn to create and customize various types of charts in Excel for data visualization and analysis.
4. Master the creation and configuration of Pivot Tables in Excel to summarize, analyze, and present data dynamically.
5. Familiarize with key Excel functionalities, terms, and their practical applications in data management and reporting.

STRUCTURE:

9.1 Data Sorting in Excel

9.1.1 Sorting by a Single Column

9.1.2 Sorting by Multiple Columns

9.2 Filtering Data in Excel

9.2.1 Single Filtering .

9.2.2 Using Multiple Filters

9.3 Charts in Excel

9.3.1 Introduction to Charts

9.3.2 Types of Charts

9.4 Pivot Tables in Excel

9.4.1 What is a Pivot Table?

9.4.2 Creating a Pivot Table

9.4.3 Configuring and Customizing Pivot Tables

9.4.4 Pivot Charts

9.4.5 Types and Use Cases

9.5 Conclusion

9.6 Key Terms

9.7 Review Questions

9.8 Suggested Readings

9.1 DATA SORTING IN EXCEL:

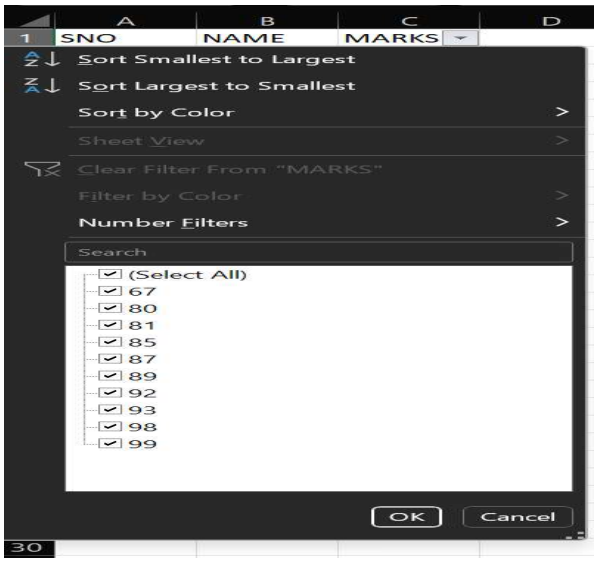
Data Sorting
When data is sorted in Microsoft Excel, rows are rearranged according to the values in a particular column. This enables you to sort numerical data, such quantities, in either ascending or descending order, or to arrange names alphabetically.

Sorting Types in Excel:
1. Sorting by a Single Column
Data can be sorted according to a single column's values.
Ex: Sorting a list of names alphabetically or numbers from least to largest are two examples.
2. Sorting by Multiple Columns
To prioritize sorting order, you can sort data according to many columns.
For instance, sorting by department first, followed by the names of the employees in each department.

9.1.1 Sorting by a Single Column
Sorting a single column in Excel is a basic activity that enables you organize data in either ascending or descending order based on the column's values.
Let's us apply Single Column Sorting in below Example
Consider the Student dataset depicted below. It has information about the S.No, Student name, Student Marks.

A	B	C
SNO	NAME	MARKS
1	Hari	89
2	Vamsi	67
3	Rahul	87
4	Kishor	98
5	Sagar	85
6	Nikhil	92
7	Kiran	80
8	Manoj	81
9	Chari	99
10	Ram	93

We will sort the Student Marks in descending order.
Select column (Marks) ->CTRL+Shift+L -> click down arrow -> select largest to smallest -> click Sort



Results will be displayed as Below

A	B	C
SNO	NAME	MARKS
9	Chari	99
4	Kishor	98
10	Ram	93
6	Nikhil	92
1	Hari	89
3	Rahul	87
5	Sagar	85
8	Manoj	81
7	Kiran	80
2	Vamsi	67

9.1.2 Sorting by Multiple Columns

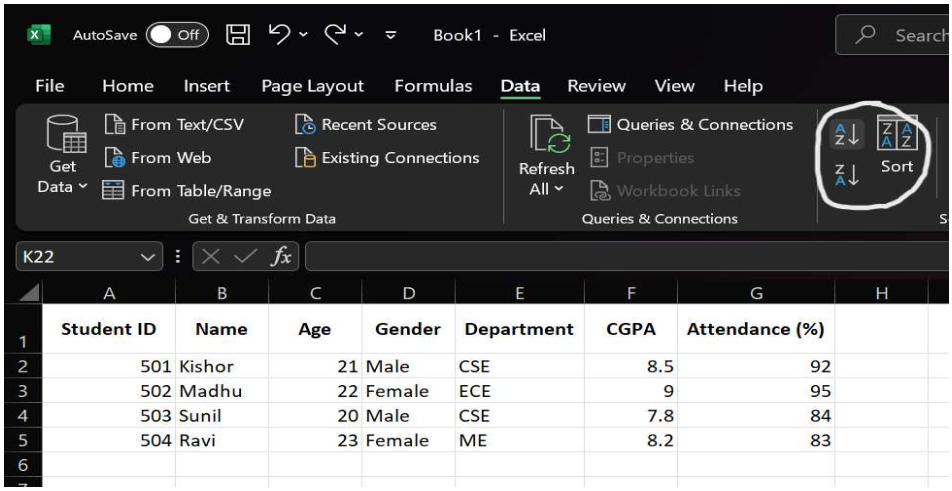
Usually, sorting just one column is required. Sometimes, though, you may need to sort over a large number of columns. Advanced sorting techniques can be used to arrange data according to many columns. Here's a detailed steps on how to accomplish it.

Consider the Following example dataset

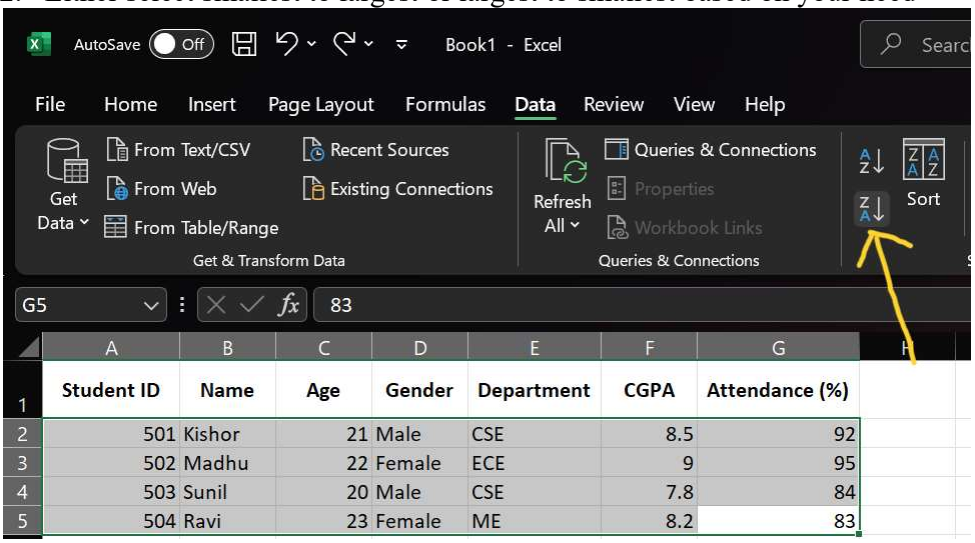
Student ID	Name	Age	Gender	Department	CGPA	Attendance (%)
501	Kishor	21	Male	CSE	8.5	92
502	Madhu	22	Female	ECE	9	95
503	Sunil	20	Male	CSE	7.8	84
504	Ravi	23	Female	ME	8.2	83

Anywhere in the data range that you wish to sort, click. Choose the full table, including any header rows, if possible.

1. Select Sort in the Data Tab



2. Either select smallest to largest or largest to smallest based on your need



3. Results will get viewed as below

Student ID	Name	Age	Gender	Department	CGPA	Attendance (%)
502	Madhu	22	Female	ECE	9	95
501	Kishor	21	Male	CSE	8.5	92
503	Sunil	20	Male	CSE	7.8	84
504	Ravi	23	Female	ME	8.2	83

9.2 FILTERING DATA IN EXCEL:

MS Excel: Filtering

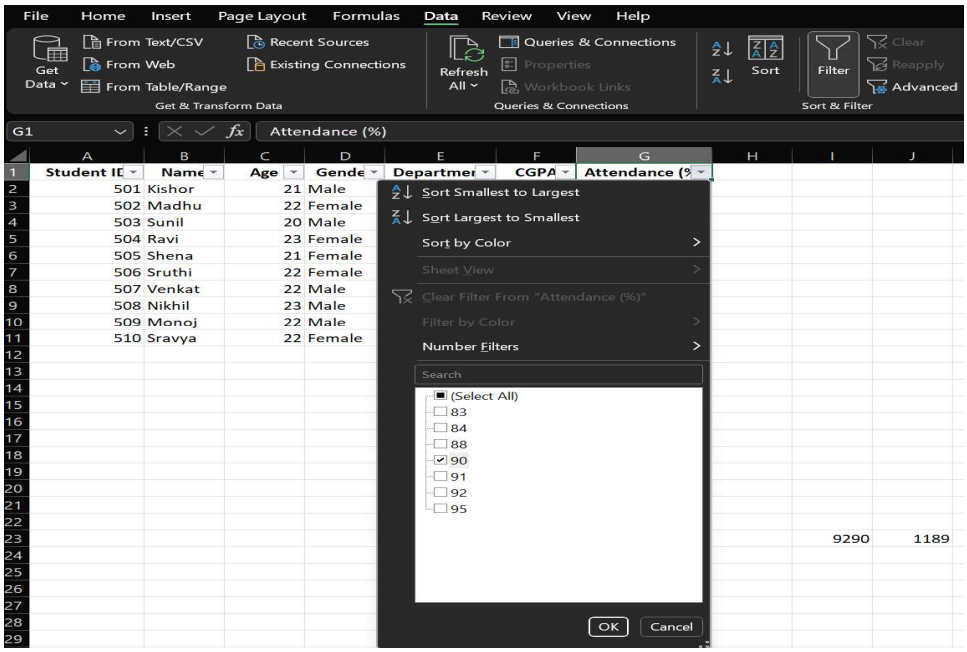
In Microsoft Excel, filtering data means showing just the rows that satisfy specific criteria. (The remaining rows are obscured or ignored)

9.2.1 Single Filtering

You can filter the data based on the single column that you wish to filter it. Consider the below example of student data

Student ID	Name	Age	Gender	Department	CGPA	Attendance (%)
501	Kishor	21	Male	CSE	8.5	92
502	Madhu	22	Female	ECE	9	95
503	Sunil	20	Male	CSE	7.8	84
504	Ravi	23	Female	ME	8.2	83
505	Shena	21	Female	EEE	8.5	92
506	Sruthi	22	Female	CSE	9	91
507	Venkat	22	Male	ECE	9	90
508	Nikhil	23	Male	CSE	7.8	91
509	Monoj	22	Male	CSE	9	88
510	Sravya	22	Female	ME	8.8	90

1. Select the column (Attendence) that you want to filter
2. Select Filter in the Data tab
3. You can select No of filters or choose any one filter based on your need



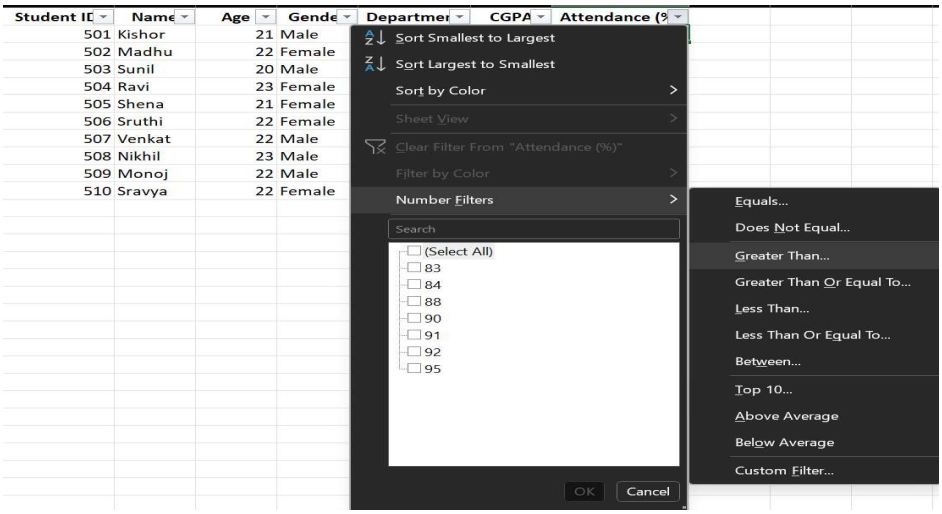
4.The Result will be shown as below

Student ID	Name	Age	Gende	Departmer	CGPA	Attendance (%
507	Venkat	22	Male	ECE	9	90
510	Sravya	22	Female	ME	8.8	90

9.2.2 Using Multiple Filters

The records can be filtered using a variety of criteria, such as different column values.
In the above example we had filtered those students who have attendance equals to 90 and now we going to filter those students who have attendance greater than 90 and belongs to CSE department

1. Filter those whose is having attendance greater than 90
   - Select the range of data from the dataset and click on filter
   - Click on attendance -> Number filters -> Greater than
   - Enter 90. Click on OK

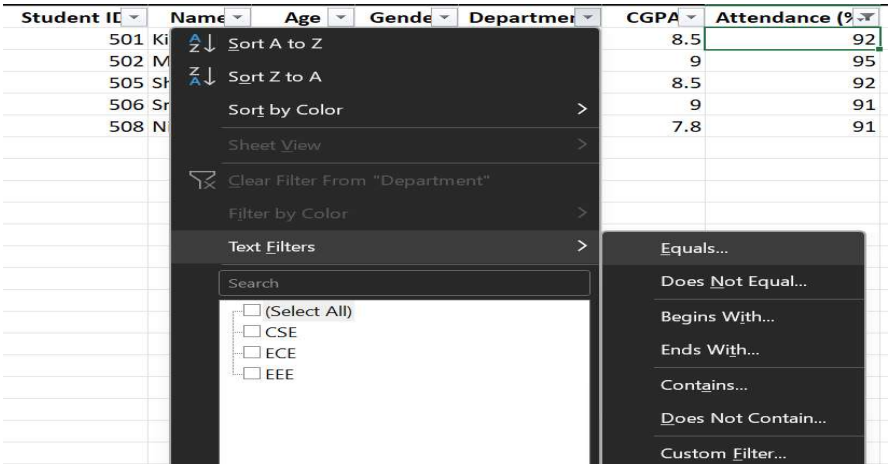


The Output will be seen as below

Student ID	Name	Age	Gende	Departmer	CGPA	Attendance (%
501	Kishor	21	Male	CSE	8.5	92
502	Madhu	22	Female	ECE	9	95
505	Shena	21	Female	EEE	8.5	92
506	Sruthi	22	Female	CSE	9	91
508	Nikhil	23	Male	CSE	7.8	91

2. Add additional filter by choosing only CSE students

- Click on Department -> Text filters -> Equals
- Enter CSE. Click on OK



The Result will be seen as below

Student ID	Name	Age	Gende	Departmer	CGPA	Attendance (%
501	Kishor	21	Male	CSE	8.5	92
506	Sruthi	22	Female	CSE	9	91
508	Nikhil	23	Male	CSE	7.8	91

9.3 CHARTS IN EXCEL:

9.3.1 Introduction to Charts

Charts in Excel offer assistance visualize information for way better understanding. Below could be a point by point step-by-step direct to making diverse sorts of charts with illustrations.

Step-1: Consider the following excel sheet

A	B	C	D
Month	**Product A**	**Product B**	**Product C**
January	500	700	600
February	600	800	750
March	750	850	900
April	900	950	1000

Step 2: Select the Data

Open Excel and enter the data.
Highlight the data range (e.g., A1:D5).

Step 3: Insert a Chart

Go to the "Insert" tab on the ribbon.
Select a Chart Type from the "Charts" group.
Choose the desired chart:
Column Chart (for comparisons)
Line Chart (for trends)
Pie Chart (for percentages)
Bar Chart (for rankings)
Scatter Chart (for relationships)
Click on the chosen chart, and it will appear in your worksheet.

Step 4: Customize the Chart

Add Chart Title: Click on the chart → Go to "Chart Elements" → Select "Chart Title" → Enter a name.
Change Colors& Styles: Click on the chart → Go to "Chart Design" → Choose a style.
Modify Axes & Labels: Click on axes → Edit text or numbers.
Add Data Labels: Right-click on the chart → Select "Add Data Labels."
Move or Resize: Drag the chart to a new position or resize it by pulling the corners.

Step 5: Save and Export the Chart

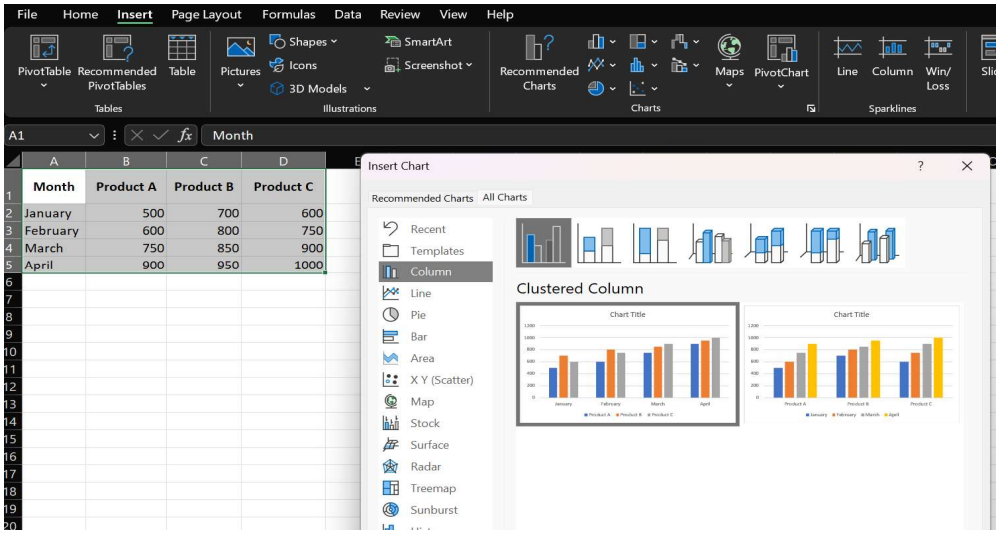
Click on the chart.
Press Ctrl + C to copy.
Paste it into PowerPoint, Word, or any document.
To save as an image:
Right-click on the chart → Select "Save as Picture" → Choose format (PNG, JPEG, etc.).

9.3.2 Types of charts

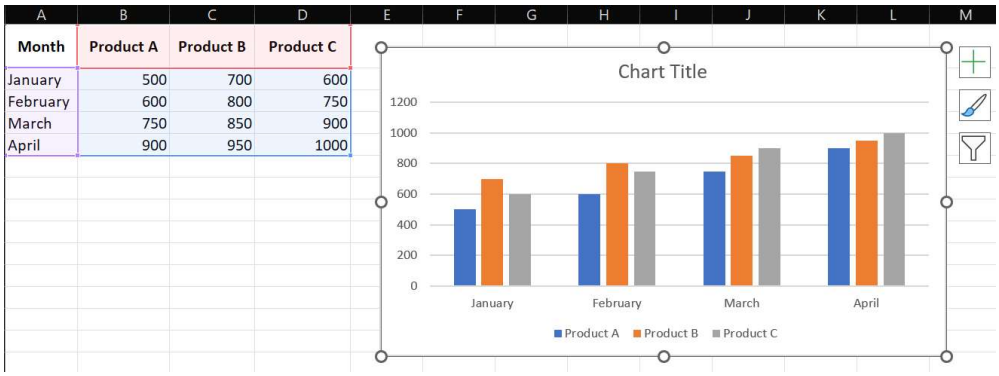
1.Column Chart

This chart is useful for comparing sales of different products over months.

In Insert tab -> Recommended chart ->all charts ->select column chart



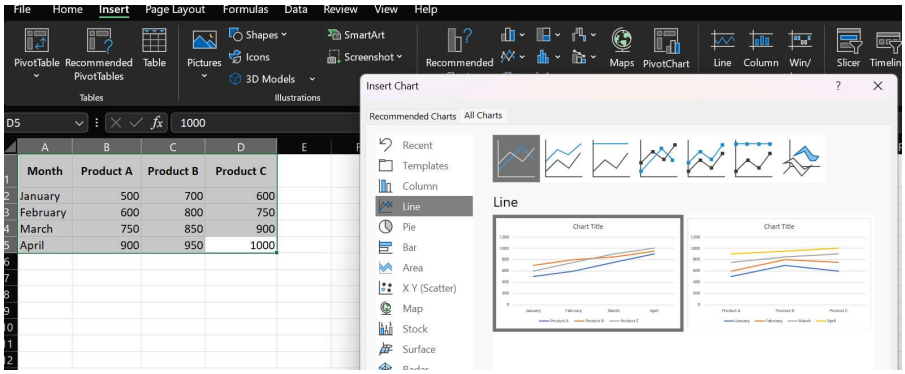
Result will be viewed as below



2. Line Chart (Trend Analysis)

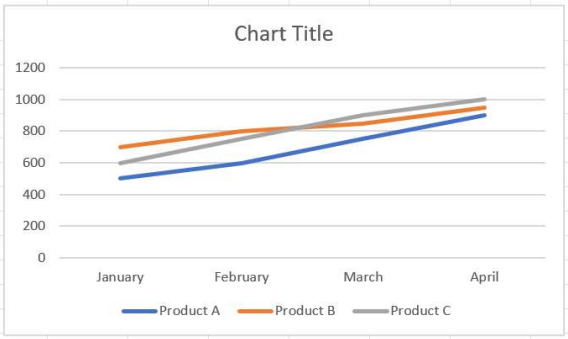
This chart helps visualize trends over time.

In Insert tab -> Recommended chart -> all charts -> select Line chart



Result will be viewed as below

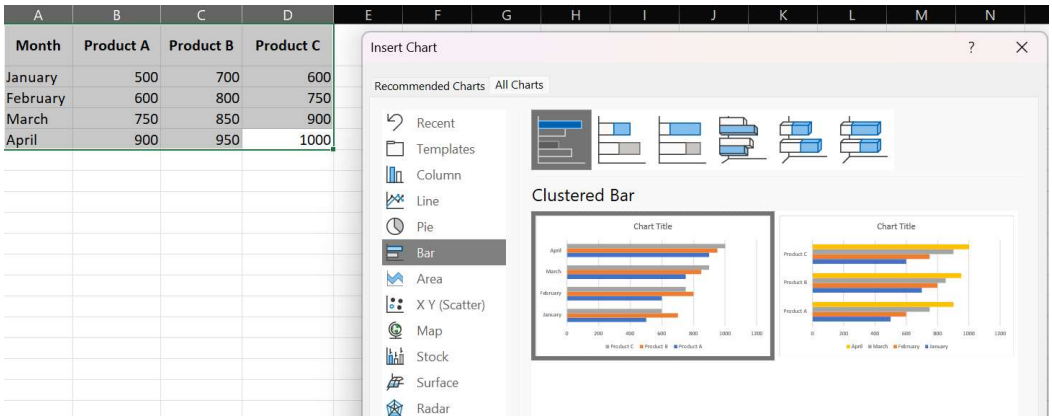
Month	Product A	Product B	Product C
January	500	700	600
February	600	800	750
March	750	850	900
April	900	950	1000



3. Bar Chart (Ranking & Comparison)

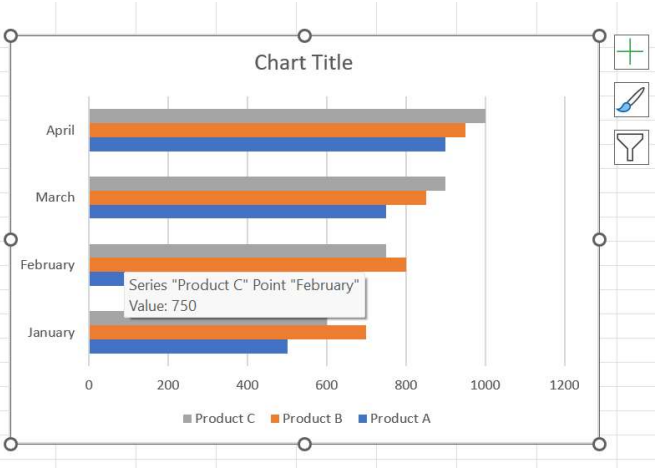
Useful when comparing different categories.

In Insert tab -> Recommended chart -> all charts -> select Bar chart



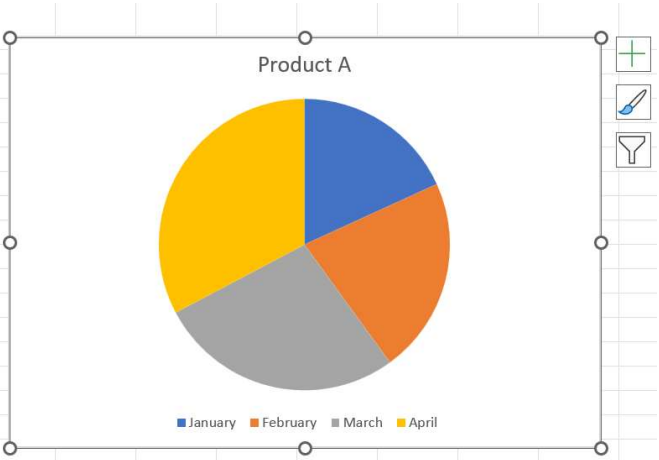
Result will be viewed as below

Month	Product A	Product B	Product C
January	500	700	600
February	600	800	750
March	750	850	900
April	900	950	1000



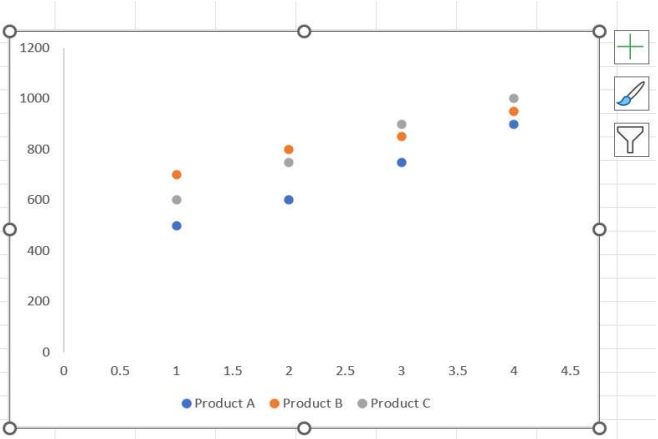
4. Pie Chart (Percentage Distribution)
This chart is useful for showing the share of each product's sales.

Month	Product A	Product B	Product C
January	500	700	600
February	600	800	750
March	750	850	900
April	900	950	1000



5. Scatter Chart (Relationship Between Two Variables)
Used for correlation between two data points.
In Insert tab -> Recommended chart -> all charts -> select Scatter chart

Month	Product A	Product B	Product C
January	500	700	600
February	600	800	750
March	750	850	900
April	900	950	1000



9.4 PIVOT TABLES IN EXCEL:

Pivot Tables
A Pivot Table in Exceed expectations may be a effective device utilized to summarize, analyze, investigate, and show data. It permits clients to reorganize and extricate valuable insights without modifying the original dataset.

9.4.1What is a Pivot Table
A Pivot Table is an intelligently way to rapidly summarize huge amounts of information. It makes a difference in:
1. Summarizing numerical information
2. Finding patterns
3. Comparing values
4. Gathering and filtering information
5. Making dynamic reports

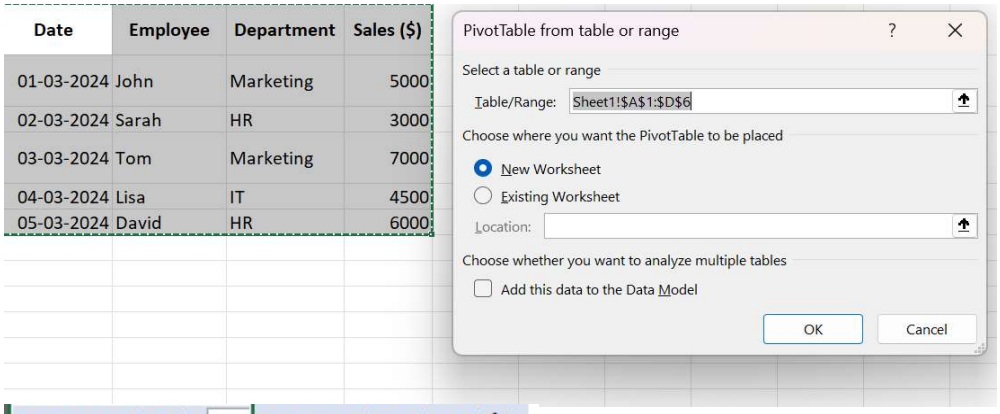
9.4.2 Creating a Pivot Table

Step -1: Consider the following table below

Date	Employee	Department	Sales ($)
01-03-2024	John	Marketing	5000
02-03-2024	Sarah	HR	3000
03-03-2024	Tom	Marketing	7000
04-03-2024	Lisa	IT	4500
05-03-2024	David	HR	6000

Step 2: Insert a Pivot Table

1. Select your dataset.
2. Go to the **Insert** tab → Click on **Pivot Table**.
3. Choose where to place the Pivot Table:
   - **New Worksheet** (recommended) or **Existing Worksheet**.
4. Click **OK**.



Row Labels	Sum of Sales ($)
01-03-2024	5000
02-03-2024	3000
03-03-2024	7000
04-03-2024	4500
05-03-2024	6000
Grand Total	**25500**

9.4.3 Step 3: Configuring and Customizing Pivot Tables

Once you insert the Pivot Table, you will see a blank table along with a **PivotTable Fields Pane** on the right side of the Excel window. This is where you will configure how your Pivot Table looks.

Understanding the PivotTable Fields Pane

This pane contains all the column names (fields) from your dataset. You can drag and drop these fields into four different areas:

Drag and Drop Fields into Areas

- Drag **"Department"** to the **Rows** section.
- Drag **"Sales ($)"** to the **Values** section.
- Excel will automatically apply the **SUM** function to Sales.

Row Labels	Sum of Sales ($)
HR	9000
IT	4500
Marketing	12000
Grand Total	**25500**

Customizing the Pivot Table

A. Sorting and Filtering

- Click on the drop-down in the **Row Labels** section.
- Sort data **Ascending/Descending** (e.g., highest to lowest sales).
- Use **Label Filters** (e.g., show only "Marketing").
- Use **Value Filters** (e.g., show departments with sales > $5000).

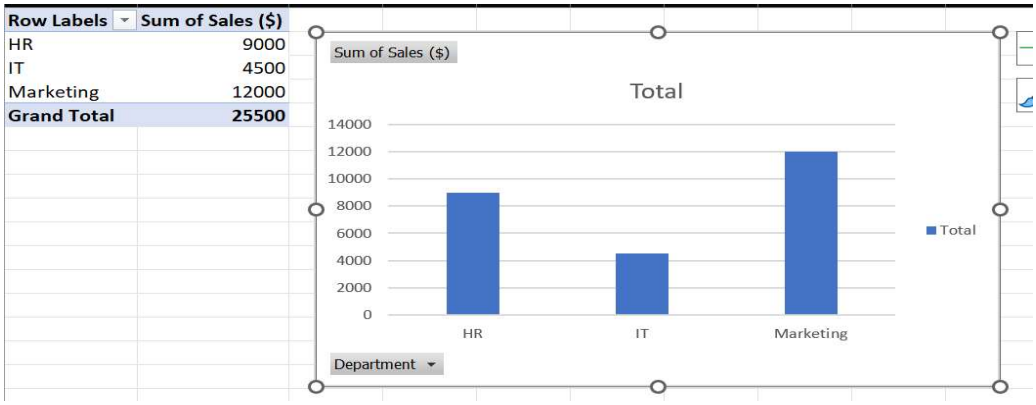
Row Labels	Sum of Sales ($)
Marketing	12000
Grand Total	**12000**

9.4.4 Step 4: Pivot Charts

To visualize the data:

1. Click anywhere inside the Pivot Table.
2. Go to **Insert → Pivot Chart**.
3. Select a chart type (Column, Pie, Line).
4. Click **OK**.

Now, your Pivot Table will be represented graphically!



Step-5: Pivot Table Types / Variations

Type	Description
Regular Pivot Table	Standard table summarizing data
Calculated Fields Pivot Table	Adds custom formulas
Dynamic Pivot Table	Updates automatically when source data changes
Two-Dimensional Pivot Table	Summarizes by both rows and columns

9.4.5 Typesand Use Cases

✅ **Sales Report** – Analyze sales by region, product, or time period.
✅ **HR Data Analysis** – Track employee headcount and department expenses.
✅ **Customer Analysis** – Summarize customer purchases and demographics.
✅ **Financial Reports** – Monitor expenses and revenue trends.

9.5 CONCLUSION:

- Summary of how sorting, filtering, charts, and Pivot Tables enhance data management in Excel.
- Emphasis on their role in transforming raw data into actionable insights.
- Preview of advanced Excel features for further learning.

9.6 KEY TERMS:

- Data Sorting, Filtering, Charts, Pivot Table, Single-Column Sorting, Multiple Filters, Data Visualization, PivotTable Fields Pane.

9.7 REVIEW QUESTIONS:

1. What is the difference between single-column and multiple-column sorting in Excel?
2. How can you filter data in Excel to show only specific rows based on multiple conditions?
3. Describe the steps to create a Pie Chart in Excel and explain when it is most useful.
4. What are the key components of a Pivot Table, and how can they be customized?
5. How does adding a Pivot Chart enhance the analysis of a Pivot Table?

9.8 SUGGESTED READINGS:

1. "Excel 2019 Bible" by Michael Alexander and Dick Kusleika – Comprehensive guide to Excel features.
2. "Microsoft Excel Data Analysis and Business Modeling" by Wayne Winston – Practical applications of Excel tools.
3. Online resources: Microsoft Excel Help (support.microsoft.com) and Excel tutorials on data analysis.

Dr. T. Srinivasa Ravi Kiran

LESSON-10

DATA TYPES, VARIABLES, AND CONSTANTS

OBJECTIVES:

1. Understand the concept of data types in C, including primary, derived, and user-defined types, and their roles in programming.
2. Explore the characteristics and usage of variables in C for storing and manipulating data.
3. Learn the definition, properties, and declaration of constants in C, including the use of #define macros.
4. Identify the syntax, rules, and practical examples for implementing data types, variables, and constants in C programs.
5. Familiarize with key C programming terms and foundational concepts to build efficient code.

STRUCTURE:

10.1 Data Types in C

10.1.1 Overview of Data Types

10.1.2 Primary Data Types

10.1.3 Derived Data Types

10.1.4 User-Defined Data Types

10.2 Variables in C

10.2.1 Definition and Purpose

10.2.2 Rules for Naming Variables

10.2.3 Usage in Programs

10.3 Constants in C

10.3.1 Overview of Constants

10.3.2 Properties of Constants

10.3.3 Constants Using #define

10.4 Conclusion

10.5 Key Terms

10.6 Review Questions

10.7 Suggested Readings

10.1 DATA TYPES IN C:

10.1.1 Overview Of Data Types

Each variable in C is linked to a particular data type, which dictates the types of data it can hold, including characters, integers, floating-point numbers, and doubles. Different data types support different operations and take up varying amounts of memory.

C data types are roughly grouped into the following types:
1. Primary Data Type: Simple values like characters, integers, and floating-point numbers are stored in these basic data types. Examples include `int`, `char`, `float`, `double`, and `void`.
2. **Derived Data Types** – These types are generated using primitive data types. Functions, pointers, and arrays are typical examples.
3. **User-Defined Data Types** – These are custom data types that programmers define as per their needs. Examples include **structures (`struct`), unions (`union`), and enumerations (`enum`).

10.1.2 Primary Data Types

C provides several built-in data types to store different types of values. The size of these data types depends on the system architecture and compiler. Below is a brief summary of commonly used data types:

1. Integer Data Type (int)

- Stores whole numbers (positive, negative, and zero).
- Can be decimal, octal, or hexadecimal.
- **Size:** 4 bytes
- **Range:** -2,147,483,648 to 2,147,483,647
- **Format Specifier:** %d
- **Variations:**
    - unsigned int (only positive values)
    - short int (2 bytes, smaller range: -32,768 to 32,767)
    - long int (larger range than int)
    - unsigned short int (similar to short int, but only positive)

example:

```
#include <stdio.h>
int main()
{
 int num = 10;
 printf("Integer value: %d", num);  // Integer value: 10
return 0;
}
```

2. Character Data Type (char)

- Stores a single character.
- **Size:** 1 byte
- **Range:** (-128 to 127) or (0 to 255)
- **Format Specifier:** %c

Example:

```
#include <stdio.h>
int main()
{
    char letter = 'A';
    printf("Character: %c", letter);   // Character: A
    return 0;
}
```

3. Floating-Point Data Type (float)

- Stores decimal numbers with single precision.
- **Size:** 4 bytes
- **Range:** 1.2E-38 to 3.4E+38
- **Format Specifier:** %f

Example:

```
#include <stdio.h>
int main() {
   float num = 3.14;
   printf("Float value: %f", num); //Float value: 3.140000
   return 0;
}
```

4. Double Data Type (double)

- Stores decimal numbers with double precision.
- **Size:** 8 bytes
- **Range:** 1.7E-308 to 1.7E+308
- **Format Specifier:** %lf

Example:

```
#include <stdio.h>
int main() {
   double num = 3.1415926535;
   printf("Double value: %lf", num); // Double value: 3.141593
   return 0;
}
```

5. Void Data Type (void)

- Represents the absence of data.
- Used in function return types, function parameters, and pointers.
- Example: void exit(int check);

```
#include <stdio.h>
void display() {
   printf("This is a void function.");
}
int main() {
   display(); // This is a void function.
   return 0;
}
```

6. Long Double (long double)

- Stores floating-point numbers with extended precision.
- **Size:** 16 bytes
- **Range:** 3.4E-4932 to 1.1E+4932
- **Format Specifier:** %Lf

Example:

```
#include <stdio.h>
int main() {
   long double num = 3.141592653589793238;
   printf("Long double value: %Lf", num); // Long double value: 3.141593
   return 0;
}
```

10.1.3Derived Data types

Primitive (built-in) data types are used in C to construct derived data types. These consist of pointers, arrays, and functions. Derived types make organized programming, memory management, and sophisticated operations possible.

1. Functions

In C, a function is a block of reusable code that carries out a particular operation. The main() function, which may call other functions, initiates execution.

Essential Elements of a Function:

1. Return Type: Indicates the kind of value that the function will yield.
2. **Function Name**: A special number that is used to invoke the function.
3. **Parameters:** Values entered into the function are called parameters.
4. **Function body:** The executable statements are located in the function body.
5. **Return Statement:** If the function is not void, it returns a value.

Example:

```
#include <stdio.h>
int addition(int a, int b) {
   return a + b;
}
int main() {
   int result = addition(5, 3);
   printf("Addition is : %d", result);   // Addition is : 15
   return 0;
}
```

2. Arrays

An array is a fixed-size collection of similar data types stored in contiguous memory locations. Arrays allow efficient data management and quick element access.

Key Properties of Arrays:

1. Fixed in size, determined at compile time.
2. Elements must be of the same type.
3. Stored in a contiguous memory block.
4. Allows random access using an index.
5. Supports multi-dimensional structures (e.g., 2D arrays).

Example:

```
#include <stdio.h>
int main() {
   int array[] = {1, 2, 3};
   for (int i = 0; i < 3; i++) {
      printf("%d ", array[i]);        // 1 2 3
   }
   return 0;
}
```

3. Functions

A pointer may be a variable that contains the memory address of another variable. It makes efficient data structures, function pointers, and dynamic memory allocation possible.

Important Pointer Properties:

1. holds a variable's or function's address.
2. Accesses stored values by using the dereference operator *.

3. Multi-level pointers, such as pointer to pointer, are supported.
4. may vary in size depending on the system.

Example:

```
#include<stdio.h >
int main() {
   int num = 20;
   int *ptr = &num;
   printf("Address stored in ptr: %p\n", ptr);   // Address stored in ptr: 0x7ffd0850df9c
   printf("Value of num: %d\n", num);          // Value of num: 20
   printf("Value using *ptr: %d\n", *ptr);     // Value using *ptr: 20
return 0; }
```

10.1.4 User Defined Data types

User-defined data types are those that are created by the user themselves. The current datatypes are the source of these datatypes.

Need of user defined data types

1. It allows the code to be customized.
2. Users are able to build more adaptable and efficient code.
3. provides abstraction.

Types of User-Defined DataTypes

There are 4 sorts of user-defined information sorts in C. They are

1. Structure
2. Union
3. Enum
4. Typedef

1.Structures

Although C lacks C++'s built-in object-oriented features, structures can be utilized to partially encapsulate code. Items of many sorts can be grouped together into a single type using structures. To define a structure, use the "struct" keyword. The structure's size is on equal with or larger than the sum of the sizes of its constituent parts.

Syntax:

```
struct structure_name {
   data_type member_name1;
   data_type member_name1;
   ....
   ....
};
```

Example

```
#include <stdio.h>
struct Person {
   char company[50];
   int lifespan;
};
int main()
{
   struct Person person1;
   strcpy(person1.company, "WelcomeToCollege");
   person1.lifespan = 30;
   printf("Name: %s\n", person1.company);   // Name: WelcomeToCollege
```

```
    printf("Age: %d\n", person1.lifespan);    // Age: 30
    return 0;
}
```

2. Union

In many respects, unions are like structures. A union differs in that only one member contains data at a time because all of the union's members are kept in the same memory address. The size of the largest member determines the size of the union. The "union" keyword is used to declare a union.

Syntax:

```
union union_name {
    datatype member1;
    datatype member2;
    ...
};
```

Example

```
#include <stdio.h>
union Data {
    int j;
    float f1;
    char str1[20];
};
int main()
{
    union Data data;
    data.j = 10;
    printf("Data.j: %d\n", data.j);
    data.f1= 3.14;
    printf("Data.f1: %f\n", data.f1);
    strcpy(data.str1, "WelcomeToCollege, C");
    printf("Data.str1: %s\n", data.str1);
    return 0;
}
```

3. Enum

"Enumeration" is shortened to "enum." With a collection of named integer constants, it enables the user to construct unique data types. An enumeration is declared with the "enum" keyword. Enum streamlines and improves the program's readability.

Syntax:

enum*enum_name* {const1, const2, ..., constN};

Here, the const1 will be alloted 0, const2 = 1, and so on within the arrangement.

4.Typedef

The current data type names can be redefined using typedef. In essence, it is employed to give the current data types new names. For this, the keyword "typedef" is utilized.

Syntax: typedef *existing_name alias_name;*

Example:

```
#include <stdio.h>
typedef char Company;
int main()
```

```
{
   Company* msg = "WelcomeToCollege";
   printf("Message: %s \n", msg);    // Message: WelcomeToCollege
   return 0;
}
```

10.2 VARIABLES IN C:

10.2.1 Definition and Purpose

A variable in C is a name for a memory region that allows us to store and retrieve data as needed. It enables us to access the location of memory without needing to commit the memory address to memory. In expressions, a variable name can be used in place of the value it holds.

Syntax: data_type variable_name;

Example:

int vari; // integer variable
char a; // character variable
float f; // float variable
data_type variable_name; = value;

10.2.2 Rules for Naming variables

As long as the variable complies with the following guidelines, we are free to give it any name:

1. The only characters allowed in a variable name are letters, numbers, and underscores.
2. A variable name must begin with either an underscore or an alphabet. It can't begin with a number.
3. The variable name cannot contain any white spaces.
4. No reserved word or keyword may be used as the name of a variable.

10.2.3 Usage in Programs

Example:

```
#include <stdio.h>
int main() {
   // Declaring variables
   int var;
   // Storing data
   var = 5;
   printf("%d", var);          // 5
   return 0;
}
```

10.3 CONSTANTS IN C:

10.3.1 Overview Of Constants

Constants are read-only values in C programming that cannot be changed while a program is running. These constants may be character, string, integer, or floating-point constants, among other sorts. They start with the declaration and don't change until the program is over.

Syntax: **co*nst*** *data_type var_name = value;*

10.3.2 Properties of Constants

1. **Initialization with Declaration**

In C, a constant variable must be initialized at the time of declaration. Otherwise, it retains the previous garbage value in memory

2. **Immutability**

Constant variables in C can only be initialized once and are immutable; their values cannot be changed afterwards.

Example:

```
#include <stdio.h>
int main() {
   const int var=100;
     printf("%d", var);   // 100
   return 0;
}
```

10.3.3 Constants Using #define

The #define directive in C generates macros, which are data-type-less symbolic constants that are updated at build time with their values.

Syntax: #**define** CONSTANT_NAME value

Example

```
#include <stdio.h>
#define PI 3.14
int main() {
   printf("%.2f", PI);       // 3.14
return 0;
}
```

10.4 CONCLUSION:

- Summary of data types, variables, and constants as core elements of C programming.
- Importance of these concepts in building structured and efficient programs.
- Preview of advanced C topics like control structures and functions for further learning.

10.5 KEY TERMS:

- Data Type, Variable, Constant, Primary Data Type, Derived Data Type, User-Defined Data Type, Pointer, Array, Structure, Union, Enum, Typedef, #define.

10.6 REVIEW QUESTIONS:

1. What are the main categories of data types in C, and how do they differ?
2. Explain the purpose of the void data type with an example of its use.
3. How do arrays and pointers enhance data management in C programs?
4. What are the key differences between a struct and a union in C?
5. Describe the rules for naming variables in C and provide an example of a valid variable name.

10.7 SUGGESTED READINGS:

1. "The C Programming Language" by Brian W. Kernighan and Dennis M. Ritchie – The definitive guide to C.
2. "C Programming Absolute Beginner's Guide" by Greg Perry and Dean Miller – A beginner-friendly introduction.
3. Online resources: C documentation on cplusplus.com and tutorials from Geeks for Geeks.

Dr. T. Srinivasa Ravi Kiran

LESSON-11

OPERATORS AND CONTROL FLOW STATEMENTS

OBJECTIVES:

1. Understand the role and types of operators in C for performing mathematical, logical, and bitwise operations on variables and values.
2. Explore the functionality of control flow statements in C to manage program execution through decision-making, looping, and jumps.
3. Learn the syntax, purpose, and practical application of various operators and control statements with examples.
4. Identify how operators and control flow statements interact to create efficient and dynamic C programs.
5. Familiarize with key terms and concepts related to operators and control flow in C programming.

STRUCTURE:

11.1 Operators in C

11.1.1 Overview of Operators

11.1.2 Types of Operators

11.2 Control Flow Statements in C

11.2.1 Overview of Control Flow

11.2.2 Conditional Statements

11.2.3 Looping Statements

11.2.4 Jump Statements

11.3 Conclusion

11.4 Key Terms

11.5 Review Questions

11.6 Suggested Readings

11.1 OPERATORS IN C:

11.1.1 Overview of Operators

The symbol that enables us to carry out particular mathematical, relational, bitwise, conditional, or logical operations on values and variables is known as an operator in C. Operands are the variables and values that are used with operators. Thus, the symbols that carry out operations on operands are known as operators.

11.1.2 Types of Operators

C language provides a wide range of operators that can be classified into 6 types based on their functionality:

1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. Bitwise Operators
5. Assignment Operators
6. Increment and Decrement Operators

1.Arithmetic Operators

These operators are used for arithmetic calculations.

Operator	Description	Example
+	Addition	a + b
-	Subtraction	a - b
*	Multiplication	a * b
/	Division	a / b
%	Modulus (Remainder)	a % b

Example:

```
#include <stdio.h>
int main()
{
   int a = 15, b = 10;
printf("a + b = %d\n", a + b);    // 25
printf("a - b = %d\n", a - b);     // 5
printf("a * b = %d\n", a * b);   // 150
printf("a / b = %d\n", a / b);    // 1
printf("a % b = %d\n", a % b);  // 5
   return 0;
}
```

2.Relational Operators

used to return true (1) or false (0) after comparing values.

Operator	Description	Example
==	Equal to	a==b
!=	Not equal to	a!=b
>	Greater than	a>b
<	Less than	a< b
>=	Greater than or equal to	a>=b
<=	Less than or equal to	a<=b

Example:

```
#include <stdio.h>
int main()
{
   int a = 10, b = 5;
```

```
printf("a <b  : %d\n", a < b);// a<b: 0
printf("a >b  : %d\n", a > b);//a>b: 1
printf("a <= b: %d\n", a <= b);  // a<=b: 0
printf("a >= b: %d\n", a >= b); //a>=b: 1
printf("a == b: %d\n", a == b);  // a==b: 1
printf("a != b : %d\n", a != b); //a!=b: 0
   return 0;
}
```

3.Logical Operators

Used to perform logical operations.

Operator	Description	Example
&&	Logical AND	(a>5) && (b<10)
||	Logical OR	(a>5) || (b<10)
!	Logical NOT	!(a>5)

Example

```
#include <stdio.h>
int main()
{
   int x = 10, y = 5;
printf("x &&y : %d\n", x&&y);      //  x&& y :1
printf("x || y : %d\n", x || y);          //   x || y : 1
printf("!x: %d\n", !x);                  //   !x : 0
   return 0;
}
```

4.Bitwise Operators

Used for bit-level operations.

Operator	Description	Example
&	Bitwise AND	a & b
!	Bitwise OR	a | b
^	Bitwise XOR	a ^ b
~	Bitwise Compliment	~a
<<	Left Shift	a << 2
>>	Right Shift	a >> 2

Example:

```
#include <stdio.h>
int main()
{
   int a = 25, b = 5;
printf("a & b: %d\n", a & b);    //  a& b: 1
printf("a | b: %d\n", a | b);       // a | b: 29
printf("a ^ b: %d\n", a ^ b);     // a ^ b: 28
printf("~a: %d\n", ~a);          // ~a: -26
printf("a >> b: %d\n", a >> b);   //   a >> b: 0
printf("a << b: %d\n", a << b);    // a << b: 800
   return 0;
}
```

5.Assignment Operators

Used to assign values to variables.

Operator	Description	Example
=	Assign	a = b
+=	Add and assign	a += b (Same as a = a + b)
^	Bitwise XOR	a -= b (Same as a = a - b)
~	Bitwise Compliment	a *= b (Same as a = a * b)
<<	Left Shift	a /= b (Same as a = a / b)
>>	Right Shift	a %= b (Same as a = a % b)

Example:

```
#include <stdio.h>
int main()
{
   int a = 25, b = 5;
printf("a = b: %d\n", a = b);          //a = b: 5
printf("a += b: %d\n", a += b);     //  a += b: 10
printf("a -= b: %d\n", a -= b);      //  a -= b: 5
printf("a *= b: %d\n", a *= b);    //  a *= b: 25
printf("a /= b: %d\n", a /= b);     //  a /= b: 5
printf("a >>= b: %d\n", a >>= b);   //  a>>= b: 0
printf("a <<= b: %d\n", a <<= b);  //  a<<= b: 0
   return 0;
}
```

6.Increment and Decrement Operators

Used to increase or decrease a variable's value.

Operator	Description	Example
++	Increment	a++ (Post-increment) or ++a (Pre-increment)
--	Decrement	a-- (Post-decrement) or --a (Pre-decrement)

Example:

```
#include <stdio.h>
int main() {
   int a = 5, b = 5;
printf("Pre-increment: ++a = %d\n", ++a);  // Output: 6
printf("Post-increment: a++ = %d\n", a++); // Output: 6
printf("After post-increment, a = %d\n", a); // Output: 7
printf("Pre-decrement: --b = %d\n", --b);  // Output: 4
printf("Post-decrement: b-- = %d\n", b--); // Output: 4
printf("After post-decrement, b = %d\n", b); // Output: 3
   return 0;
}
```

11.2 Control Flow Statements in C

11.2.1 Overview Of Control Flow

The order in which statements in a program are executed is determined by control flow statements in C. By controlling loops, jumps, and decision-making, these statements give programmers effective control over how their programs are executed.

11.2.2 Conditional Statements

Types of Control Statements in C:

1. Conditional Statements
2. Looping Statements
3. Jump Statements

1.Conditional Statement

These statements allow a program to choose between multiple paths of execution.

Statement	Description
if	Executes a block of code if the condition is true.
if-else	Executes one block if the condition is true, otherwise another block.
nested if	An if statement inside another if.
else-if ladder	Multiple conditions are checked sequentially.
switch	Allows multiple case selections based on a variable's value.

1.1 if Statement

- The if statement executes a block of code if the condition evaluates to true.

Syntax:

```
if (condition) {
   // code
}
```

Example:

```
#include <stdio.h>
int main() {
   int num = 10;
   if (num> 0) {
printf("The number is positive.\n");    // The number is positive.
   }
   return 0;
}
```

1.2 if-else Statement

- The if-else statement executes one block if the condition is true, otherwise another block.

Syntax:

```
if (condition) {
   // Code
} else {
   // Code
}
```

Example:

```
#include <stdio.h>
```

```
int main() {
   int num = -5;
   if (num> 0) {
printf("The number is positive.\n");
   } else {
printf("The number is negative.\n");        // The number is negative.
   }
   return 0;
}
```

1.3 else-if Ladder

- Used to check **multiple conditions sequentially**.

Syntax:

```
if (condition1) {
   // Executes if condition1 is true
} else if (condition2) {
   // Executes if condition2 is true
} else {
   // Executes if none of the conditions are true
}
```

Example:

```
#include <stdio.h>
int main() {
   int num = 0;
   if (num> 0) {
printf("Positive number\n");
   } else if (num< 0) {
printf("Negative number\n");
} else {
printf("Zero\n");          // Zero
   }
   return 0;
}
```

1.4 switch Statement

- Used **instead of multiple if-else** statements for better readability.

Syntax:

```
switch (variable) {
   case value1:
      // Code for value1
      break;
   case value2:
      // Code for value2
      break;
   default:
      // Code if no case matches
}
```

Example:

```
#include <stdio.h>
int main() {
```

```
    int day = 3;
    switch (day) {
        case 1:
printf("Monday\n");
            break;
        case 2:
printf("Tuesday\n");
            break;
        case 3:
printf("Wednesday\n");        // Wednesday
            break;
        default:
printf("Invalid day\n");
    }
    return 0;
}
```

11.2.3 Looping Statements

Loops execute a block of code multiple times based on a condition.

2.1 for Loop

• Executes a piece of code for a settled number of times.

Syntax:

```
for (initialization; condition; update) {
    // Code
}
```

Example:

```
#include <stdio.h>
int main() {
    for (int i = 1; i<= 5; i++) {
printf("%d ", i);              //   1 2 3 4 5
    }
    return 0;
}
```

2.2 while Loop

- Executes **as long as the condition is true**.

Syntax:

```
while (condition) {
    // Code to execute
}
```

Example:

```
#include <stdio.h>
int main() {
    int i = 1;
    while (i<= 5) {
printf("%d ", i);
i++;
    }
    return 0;
```

```
}
```
Output:
1 2 3 4 5

2.3 do-while Loop
- **Executes at least once**, even if the condition is false.

Syntax:
```
do {
   // Code to execute
} while (condition);
```
Example:
```
#include <stdio.h>
int main() {
   int i = 1;
   do {
printf("%d ", i);
i++;
   } while (i<= 5);

   return 0;
}
```
Output:
1 2 3 4 5

11.2.4. Jump Statements
Jump statements **alter the normal flow** of execution.

3.1 break Statement
- **Exits a loop or switch statement** immediately.

Example:
```
#include <stdio.h>
int main() {
   for (int i = 1; i<= 10; i++) {
      if (i == 5) {
break;  // Stops when i = 5
      }
printf("%d ", i);
   }
   return 0;
}
```
Output:
1 2 3 4

3.2 continue Statement
- **Skips the current iteration** and moves to the next.

Example:
```
#include <stdio.h>
int main() {
   for (int i = 1; i<= 5; i++) {
      if (i == 3) {
```

```
continue;  // Skips when i = 3
        }
printf("%d ", i);
    }
    return 0;
}
```

Output:

1 2 4 5

3.3 goto Statement

- **Jumps to a labeled statement**.

Example:

```
#include <stdio.h>
int main() {
    int num = 1;
    if (num == 1) {
goto label;
    }
printf("This will be skipped\n");
label:
printf("Jumped to label\n");
    return 0;
}
```

Output:

Jumped to label

3.4 return Statement

- **Exits a function** and optionally returns a value.

Example:

```
#include <stdio.h>
int sum(int a, int b) {
    return a + b;  // Returns sum of a and b
}

int main() {
    int result = sum(5, 7);
printf("Sum: %d\n", result);
    return 0;
}
```

Output:

Sum: 12

11.3 CONCLUSION:

- Summary of how operators and control flow statements work together to enable logic and repetition in C programs.
- Emphasis on their role in creating flexible and efficient code.
- Preview of advanced topics like functions and pointers for deeper exploration.

11.4 KEY TERMS:

- Operator, Operand, Arithmetic Operator, Relational Operator, Logical Operator, Bitwise Operator, Assignment Operator, Increment/Decrement Operator, Control Flow, Conditional Statement, Looping Statement, Jump Statement, if, switch, for, while, do-while, break, continue, go to, return.

11.5 REVIEW QUESTIONS:

1. What are the different types of operators in C, and how do they differ in functionality?
2. How do relational operators differ from logical operators, and when would you use each?
3. Explain the difference between pre-increment (++a) and post-increment (a++) with an example.
4. How does a switch statement improve readability compared to multiple if-else statements?
5. What is the key difference between a while loop and a do-while loop, and when would you use each?

11.6 SUGGESTED READINGS:

1. "The C Programming Language" by Brian W. Kernighan and Dennis M. Ritchie – Foundational text on C.
2. "C Programming: A Modern Approach" by K.N. King – Detailed coverage of operators and control structures.
3. Online resources: Tutorials from Learn-C.org and C programming documentation on cplusplus.com.

Dr. T. Srinivasa Ravi Kiran

LESSON-12

FUNCTIONS AND RECURSION

OBJECTIVES:

1. Understand the concept of functions in C, including their structure, types, and role in promoting modularity and code reusability.
2. Explore the mechanisms of function declaration, definition, and calling, along with return types and argument passing techniques.
3. Learn the concept of recursion in C, including its working principles, types, and practical applications.
4. Identify the advantages and disadvantages of using functions and recursion in programming, with examples.
5. Familiarize with key terms and syntax related to functions and recursion to develop efficient C programs.

STRUCTURE:

12.1 Functions in C

- **12.1.1 Overview of Functions**
- **12.1.2 Structure of Functions**
- **12.1.3 Function Components**
- **12.1.4 Types of Functions**
- **12.1.5 Passing Arguments**
- **12.1.6 Advantages and Disadvantages**

12.2 Recursion in C

- **12.2.1 Overview of Recursion**
- **12.2.2 How Recursion Works**
- **12.2.3 Types of Recursion**
- **12.2.4 Advantages and Disadvantages**
- **12.2.5 When to Use Recursion**

12.3 Conclusion

12.4 Key Terms

12.5 Review Questions

12.6 Suggested Readings

12.1 FUNCTIONS IN C:

12.1.1 Overview of Functions

A function in C could be a set of articulations that when called perform a few particular errands. It is the essential building piece of a C program that gives measured quality and code

reusability. The programming explanations of a work are encased inside {} braces, having certain implications and performing certain operations. They are too called subroutines or methods in other dialects.

12.1.2 Structure of Functions

1. Function Declaration
2. Function Definition
3. Function Calls

Function Declaration

In a function announcement, we must give the function title, its return type, and the number and type of its parameters. A function statement tells the compiler that there's a work with the given title defined somewhere else within the program.

Syntax: return_type function_name (parameter_1, parameter_2);
Example: int multiplication(int *x*, int *y*); // Function declaration with parameter names

Function Definition

The function definition comprises of real explanations which are executed when the function is called (i.e. when the controller reaches the function).
C function is for the most part characterized and pronounced in a single step since the function definition continuously begins with the function declaration so we don't ought to announce it explicitly. The below illustration serves as both a function definition and a declaration.
Syntax:

return_type**function_name** (param1_type *param1_name,* param2_type *param2_name*)
{
// function body
}

Function Call

A function call may be a statement that instructs the compiler to execute the function. We use the function title and parameters within the function call.
Within the below case, the primary sum function is called and 10,30 are passed to the sum function. After the function call sum of a and b is returned and control is additionally returned back to the main function of the program.
Example:

```
#include <stdio.h>
int sum(int x, int y)   <---------------------------- Function Definition
{
  return x + y;
}
int main()
{
  int add = sum(10, 30);   <---------------------------------- Function Calling
printf("Sum is: %d", add);       // Sum is: 40
  return 0;
}
```

12.1.3 Function Components

Function Return Type

Function return type tells what kind of value is returned after all function is executed. When we do not need to return a value, we can use the void data type.

Example: int function(parameter1,parameter2);

Function Arguments

Function arguments, also called function parameters, refer to the data provided to a function as input to process and execute tasks.

Example: int function_name(int para1, int para2);

Different types of Return types and Arguments

1. Function with no arguments and no return value
2. Function with no arguments and with return value
3. Function with argument and with no return value
4. Function with arguments and with return value

12.1.4 Types of Functions

1. Inbuilt Functions
2. User defined functions

Inbuilt Functions

A library function is additionally referred to as a "built-in function". A compiler package as of now exists that contains these functions, each of which encompasses a particular meaning and is included within the package. Built-in capacities have the advantage of being straightforwardly usable without being characterized, though user-defined capacities must be pronounced and characterized some time recently being utilized.

Example: pow(), sqrt(), strcmp(), strcpy() etc.

Example usage:

```
#include <math.h>
#include <stdio.h>
int main()
{
  double num;
num = 64;
  double sqRt = sqrt(num);
printf("The Square root of %.2lf = %.2lf", num, sqRt);   // The Square root of 64.00 = 8.00
  return 0;
}
```

User Defined Function

Functions that the programmer makes are known as User-Defined functions or "tailor-made functions". User-defined functions can be progressed and altered concurring to the require of the programmer. At whatever point we compose a function that's case-specific and isn't defined in any header file, we need to announce and characterize our claim functions concurring to the syntax.

Example:

```
#include <stdio.h>
int sum(int x, int y)
{
```

```
  return x + y;
}
int main()
{
  int a = 20, b = 70;
  int res = sum(a, b);
printf("Sum is: %d", res);         // Sum is: 90
  return 0;
}
```

12.1.5 Passing Arguments

1. Pass by Value
2. Pass by Reference

Pass by Value

Parameter passing in this strategy duplicates values from genuine parameters into formal work parameters. As a result, any changes made interior the functions don't reflect within the caller's parameters.

Example:

```
#include <stdio.h>
void swap(int var1, int var2)
{
  int temp = var1;
  var1 = var2;
  var2 = temp;
}
int main()
{
  int var1 = 4, var2 = 5;
printf("Before swap Value of var1 and var2 is: %d, %d\n", var1, var2);
swap(var1, var2);
printf("After swap Value of var1 and var2 is: %d, %d",var1, var2);
  return 0;
}
```

Output: Before swap Value of var1 and var2 is: 4, 5
After swap Value of var1 and var2 is: 5, 4

Pass by Reference

The caller's actual parameters and the function's actual parameters refer to the same areas, so any changes made interior the work are reflected within the caller's actual parameters.

Example:

```
#include <stdio.h>
void swap(int *var1, int *var2)
{
  int temp = *var1;
  *var1 = *var2;
  *var2 = temp;
}
int main()
{
  int var1 = 5, var2 = 15;
```

```
printf("Before swap Value of var1 and var2 is: %d, %d\n", var1, var2);
swap(&var1, &var2);
printf("After swap Value of var1 and var2 is: %d, %d",var1, var2);
  return 0;
}
```

Output:

Before swap Value of var1 and var2 is: 5, 15
After swap Value of var1 and var2 is: 15, 5

12.1.6 Advantages of Functions

- The function can reduce the repetition of the same explanations within the program.
- The function makes code clear by giving modularity to our program.
- There's no fixed number of calling functions it can be called as numerous times as you need.
- The function decreases the size of the program.
- Once the function is pronounced you'll fair utilize it without considering around the inside working of the work.

Disadvantages of Functions

- Cannot return different values.
- Memory and time overhead due to stack outline assignment and exchange of program control.

12.2 RECURSION IN C:

12.2.1 Overview Of Recursion

Recursion could be a programming procedure where a work calls itself to solve a issue. It is especially valuable for understanding issues that can be broken down into littler, comparative subproblems.

12.2.2 How Recursion Works?

A recursive work in C takes after two fundamental principles:

1. **Base Case:**

Usually the condition that stops the recursion. Without a base case, recursion would proceed uncertainly, driving to stack overflow.

2. **Recursive Case:**

This characterizes the function's call to itself with a altered contention, gradually lessening the problem's complexity.

Syntax:

```
return_typefunction_name(parameters) {
   if (base_condition) {
      return base_value; // Base case to terminate recursion
   }
   else {
      return function_name(modified_parameters); // Recursive call
   }
}
```

12.2.3 Types of Recursion in C

1. Direct Recursion

A function straightforwardly calls itself inside its body.

Example: Factorial calculation.

```
#include <stdio.h>
int fact(int n) {
   if (n == 0)
      return 1; // Base case
   else
      return n * fact(n - 1); // Recursive call
}
int main() {
   int num = 5;
printf("Factorial value of  %d is %d", num, fact(num));
   return 0;
}
```

Output: Factorial value of 5 is 120

2. Indirect Recursion

A function calls another function, which at that point calls the initial function.

Example: Two functions calling each other then again.

```
#include <stdio.h>
void function1(int n);
void function2(int n);
void function1(int n) {
   if (n > 0) {
printf("%d ", n);
      function2(n - 1);    // function1 calling function2
   }
}
void function2(int n) {
   if (n > 1) {
printf("%d ", n);
      function1(n - 2);   // function2 calling function1
   }
}
int main() {
   function1(10);
   return 0;
}
```

Output: 10 9 7 6 4 3 1

3. Tail Recursion

The recursive call is the final statement executed some time recently returning a value.

Example: Optimized adaptation of factorial calculation.

```
#include <stdio.h>
int fact_helper(int n, int result) {
   if (n == 0)
      return result;
   return fact_helper(n - 1, n * result);
```

```
}
int factorial(int n) {
   return fact_helper(n, 1);
}
int main() {
   int num = 5;
printf("Factorial value of %d is %d", num, factorial(num));
   return 0;
}
```

Output: Factorial value of 5 is 120

4. Non-Tail Recursion

The recursive call isn't the final operation some time recently returning.

Case: Fibonacci arrangement calculation.

```
#include <stdio.h>
int fibonacci(int n) {
   if (n == 0) return 0;
   if (n == 1) return 1;
   return fibonacci(n - 1) + fibonacci(n - 2);
}
int main() {
   int num = 6;
printf("Fibonacci value at pos %d is %d", num, fibonacci(num));
   return 0;
}
```

Output : Fibonacci value at pos 6 is 8

12.2.4 Advantages and Disadvantages

✓ Makes code brief and less demanding to get it.

✓ Valuable for issues like tree traversal, chart traversal, and backtracking.

✓ Dispenses with the require for complex circles.

Impediments of Recursion

✗ Employments more memory due to operate call stack.

✗ Can cause stack flood for profound recursion.

✗ May be less productive than iterative arrangements in a few cases.

12.2.5 When to Utilize Recursion?

- When the issue includes a normal recursive structure (e.g., factorial, Fibonacci, Tower of Hanoi).
- When breaking down issues into littler subproblems disentangles usage.
- When recursion progresses code coherence and practicality.

12.3 CONCLUSION:

- Summary of functions and recursion as tools for structuring and solving problems in C.
- Emphasis on their combined utility in creating efficient, readable programs.
- Preview of advanced topics like arrays, pointers, and data structures for further learning.

12.4 KEY TERMS:

- Function, Function Declaration, Function Definition, Function Call, Return Type, Argument, Parameter, Inbuilt Function, User-Defined Function, Pass by Value, Pass by Reference, Recursion, Base Case, Recursive Case, Direct Recursion, Indirect Recursion, Tail Recursion, Non-Tail Recursion.

12.5 REVIEW QUESTIONS:

1. What are the three main components of a function in C, and how do they work together?
2. How does a user-defined function differ from an inbuilt function, and when would you use each?
3. Explain the difference between pass by value and pass by reference with an example.
4. What is the role of a base case in recursion, and what happens if it's missing?
5. Compare tail recursion and non-tail recursion, providing an example of each.

12.6 SUGGESTED READINGS:

1. "The C Programming Language" by Brian W. Kernighan and Dennis M. Ritchie – Core reference for C functions.
2. "C Programming Absolute Beginner's Guide" by Greg Perry and Dean Miller – Beginner-friendly explanations.
3. Online resources: Tutorials from Geeks for Geeks and C documentation on cplusplus.com.

Dr. T. Srinivasa Ravi Kiran

LESSON-13
ARRAYS

OBJECTIVES:

The objectives of the lesson are

1. Arrays store multiple values of the same type in contiguous memory, allowing efficient data access and manipulation. They can be one-dimensional, two-dimensional, or multi-dimensional.
2. Arrays require a declared data type and size. Initialization can happen during declaration or later. Elements are accessed using zero-based indexing.
3. Arrays can be one-dimensional, two-dimensional (matrices), or multi-dimensional. Common operations include modification, traversal, and computations, making them useful for data processing and scientific applications.

STRUCTURE:

13.1 Introduction to Arrays

- **13.1.1Arrays**
- **13.1.2. What is Array in C**
- **13.1.3. C Array Declaration**
- **13.1.4. Array Initialization with Declaration without Size**
- **13.1.5. Array Initialization after Declaration (Using Loops)**
- **13.1.6. Accessing Array Elements in C**
- **13.1.7. C Array Traversal**
- **13.1.8. Using Arrays in C**

13.2 Types of Arrays in C

- **13.2.1. One-Dimensional Array**
- **13.2.2. Two-Dimensional Array (Matrix)**
- **13.2.3. Multi-Dimensional Array (3D Array)**

13.3 Character Array (String)

13.4 String Functions in C

13.5 Array of Pointers in C

13.6 Array of Pointers to Strings (String Array)

13.7 Dynamic Array (Using malloc)

13.8 Summary Table for arrays

13.9 Summary

13.10 Key Terms

13.11 Review Questions

13.12 Suggested Readings

13.1 INTRODUCTION TO ARRAYS:

An array is a fundamental data structure in C that allows storing multiple values of the same data type under a single variable name. Arrays provide an efficient way to manage and manipulate large amounts of data by enabling indexed access to elements stored in contiguous memory locations.

Instead of declaring multiple variables separately, an array helps in grouping similar elements together, making the code more organized and reducing redundancy. Arrays can be one-dimensional (1D), two-dimensional (2D), or multi-dimensional (3D, etc.), allowing various applications such as mathematical computations, data storage, and matrix operations.

In this we will explore thedeclaration, initialization, types, syntax, advantages, and disadvantages of arrays, along with practical examples to understand their implementation.

13.1.1Arrays

Arrays in C are among the most commonly used data structures in C programming. They provide a simple and efficient way to store multiple values under a single variable name.

13.1.2. What is Array in C

In C, an array is a fixed-size collection of similar data elements stored in consecutive memory locations. It can hold primitive data types like int, char, and float, as well as derived and user-defined types such as pointers and structures.

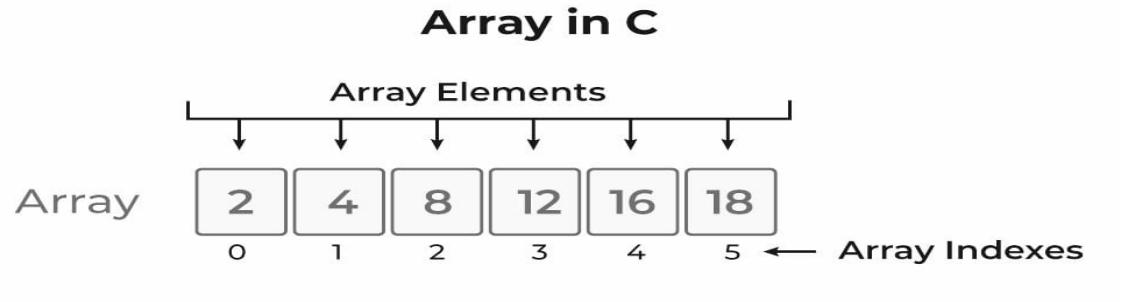


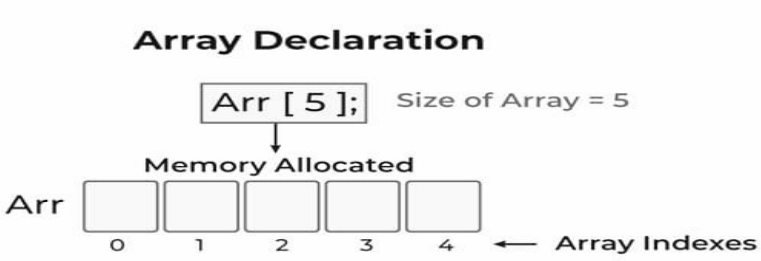
Figure: Array in C.

13.1.3. C Array Declaration

In C, an array must be declared before use, just like any other variable. The declaration includes the array name, the data type of its elements, and its size. When an array is declared, the compiler allocates a memory block of the specified size to store its elements.

Syntax of Array Declaration

data_typearray_name[size];
or
data_typearray_name[size1][size2]...[sizeN];
where N is the number of dimensions.



13.1.4. Array Initialization with Declaration without Size

When we initialize an array using an initializer list, we don't need to explicitly declare the size of the array. The compiler can automatically determine the size based on the number of elements in the initializer list.

data_typearray_name[] = {1,2,3,4,5};

The size of the array above is 5, which is automatically determined by the compiler.

13.1.5. Array Initialization after Declaration (Using Loops)

In this method, we initialize the array after declaring it by individually assigning values to each element. This can be done using a for loop, while loop, or do-while loop to assign values to the array elements.

```
for (int i = 0; i< N; i++)
{
array_name[i] = valuei;
}
```

Example of Array Initialization in C

```
#include <stdio.h>
int main() {
   // Array initialization using an initializer list
   int arr[] = {10, 20, 30, 40, 50};
   // Array initialization using an initializer list without specifying size
   int arr1[] = {1, 2, 3, 4, 5};
   // Array initialization using a for loop
   float arr2[5];
   for (int i = 0; i< 5; i++) {
      arr2[i] = i * 2.1f;
   }
   // Printing each array
printf("Integer Array 1: ");
   for (int i = 0; i< 5; i++) {
printf("%d ", arr[i]);
   }
printf("\n");
printf("Integer Array 2: ");
   for (int i = 0; i< 5; i++) {
printf("%d ", arr1[i]);
   }
printf("\n");
printf("Float Array: ");
   for (int i = 0; i< 5; i++) {
printf("%.2f ", arr2[i]);
   }
```

```
printf("\n");
   return 0;
}
```

Output

Integer Array 1: 10 20 30 40 50
Integer Array 2: 1 2 3 4 5
Float Array: 0.00 2.10 4.20 6.30 8.40

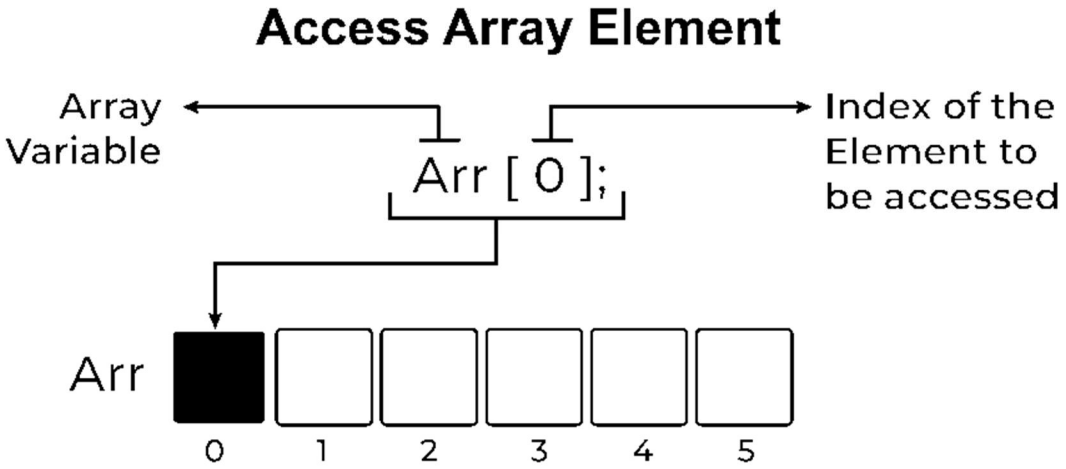
13.1.6. Accessing Array Elements in C

In C, you can access elements of an array using the array subscript operator**[ ]** along with the index value of the element.

Syntax

array_name[index];

The indexing in arrays starts from 0.
The first element is at index 0.
The last element is at index N-1, where N is the total number of elements in the array.



Example: Accessing Array Elements

```
#include <stdio.h>
int main() {
   // Define an array with 5 elements
   int numbers[] = {10, 20, 30, 40, 50};
   // Accessing elements using their index
printf("First element: %d\n", numbers[0]);  // Access element at index 0
printf("Third element: %d\n", numbers[2]);  // Access element at index 2
printf("Last element: %d\n", numbers[4]);   // Access element at index 4
   return 0;
}
```

Output:

First element: 10
Third element: 30
Last element: 50

Accessing elements beyond the array size (out of bounds) can cause undefined behavior.
Indexing starts from 0 and goes up to N-1 (not N).

You can also modify elements using indexing:
numbers[1] = 25; // Changing the second element (index 1) to 25
We can update the value of an element at the given index i in a similar way to accessing an element by using the array subscript operator **[ ]**and assignment operator **=**.
array_name[i] = new_value;

13.1.7. C Array Traversal

Traversal refers to the process of visiting each element in a data structure. In C, we use loops to iterate through every element of an array efficiently.

Traversing an Array Using a for Loop

To access each element of an array, we use a for loop:

Syntax:

```
for (int i = 0; i< N; i++) {
array_name[i]; // Access each element
}
```

Example: Traversing an Array in C

```
#include <stdio.h>
int main() {
   // Define an array with 5 elements
   int numbers[] = {10, 20, 30, 40, 50};
   int size = sizeof(numbers) / sizeof(numbers[0]); // Calculate array size
   // Traverse the array using a for loop
printf("Array Elements: ");
   for (int i = 0; i< size; i++) {
printf("%d ", numbers[i]); // Print each element
   }
printf("\n");
   return 0;
}
```

Output:

Array Elements: 10 20 30 40 50

A loop iterates from 0to N-1 (since indexing starts from 0). sizeof(array) / sizeof(array[0]) helps determine the array size dynamically. Can be used with while or do-while loops as well.

13.1.8. Using Arrays in C

Arrays in C allow us to store multiple values of the same type in a single variable. Below is a program demonstrating how to declare, initialize, modify, and traverse an array.

Example: Working with Arrays in C

```
// C Program to demonstrate array usage
#include <stdio.h>
int main() {
   // Declare and initialize an array
   int arr[5] = {10, 20, 30, 40, 50};
   // Modify the element at index 2
arr[2] = 100;
   // Traverse and print the array
printf("Elements in Array: ");
   for (int i = 0; i< 5; i++) {
printf("%d ", arr[i]);
```

```
    }
printf("\n");
    return 0;
}
```

Output:

Elements in Array: 10 20 100 40 50
Array Declaration & Initialization – int arr[5] = {10, 20, 30, 40, 50};
Modifying an Element – arr[2] = 100; (changes 30 to 100)
Array Traversal – Using a for loop to access and print elements

13.2. TYPES OF ARRAYS IN C:

Arrays in C can be categorized into different types based on their structure and usage. Below are the main types of arrays along with descriptions, example programs, and outputs.

13.2.1. One-Dimensional Array

A one-dimensional array is a collection of elements of the same data type stored in contiguous memory locations. It allows storing multiple values under a single variable name, with each value accessible using an index.

Syntax of One-Dimensional Arrays:

data_typearray_name[size];
data_type → Type of elements stored in the array (e.g., int, float, char).
array_name → Name of the array.
size → Number of elements the array can hold.

Examples:

1. Declaration and Initialization:

```
int numbers[5] = {10, 20, 30, 40, 50};  // Declares and initializes the array
```

This creates an array with 5 integer elements:

Index	Value
0	10
1	20
2	30
3	40
4	50

2.Accessing Elements

```
printf("%d", numbers[2]);  // Output: 30 (Element at index 2)
```

3. Modifying an Element:

```
numbers[1] = 25;  // Changes the second element from 20 to 25
```

4. Traversing an Array Using a Loop:

```
#include <stdio.h>
int main() {
int numbers[5] = {10, 20, 30, 40, 50};
printf("Array Elements: ");
    for (int i = 0; i< 5; i++) {
printf("%d ", numbers[i]);
    }
    return 0;
}
```

Output:
Array Elements: 10 20 30 40 50
Key Points:
Indexing starts from 0 and ends at size - 1.
Access elements using array_name[index].
Arrays in C have a fixed size (cannot grow dynamically).
Elements are stored in contiguous memory locations.

```
#include <stdio.h>
int main() {
   // Declare and initialize a one-dimensional array
   int arr[5] = {10, 20, 30, 40, 50};
   // Print elements using a loop
printf("One-Dimensional Array Elements: ");
   for (int i = 0; i< 5; i++) {
printf("%d ", arr[i]);
   }
printf("\n");
   return 0;
}
```

Output:
One-Dimensional Array Elements: 10 20 30 40 50

13.2.2. Two-Dimensional Array (Matrix)

A two-dimensional array in C is an array of arrays, where data is stored in a tabular form (rows and columns). It is commonly used to represent matrices, tables, or grids.
Syntax of a Two-Dimensional Array:
data_typearray_name[rows][columns];
data_type → The type of data stored in the array (e.g., int, float, char).
array_name → The name of the array.
rows → Number of rows in the array.
columns → Number of columns in the array.
Examples of Two-Dimensional Arrays
1. **Declaration and Initialization:**

```
int matrix[2][3] = {
   {1, 2, 3},
   {4, 5, 6}
};
```

This creates a 2x3 matrix:

Row \ Column	0	1	2
0	1	2	3
1	4	5	6

2. Accessing Elements:
Elements can be accessed using their row and column indexes.

```
printf("%d", matrix[1][2]);  // Output: 6 (Element at row 1, column 2)
```

3. Modifying an Element:

```
matrix[0][1] = 10;  // Changes the value at row 0, column 1 from 2 to 10
```

4. Traversing a 2D Array Using Nested Loops:

```
#include <stdio.h>
int main() {
   int matrix[2][3] = {
```

```
        {1, 2, 3},
        {4, 5, 6}
    };
    // Printing the 2D array
printf("Two-Dimensional Array:\n");
    for (int i = 0; i< 2; i++) {  // Loop for rows
        for (int j = 0; j < 3; j++) {  // Loop for columns
printf("%d ", matrix[i][j]);
        }
printf("\n");  // New line after each row
    }
    return 0;
}
```

Output:

Two-Dimensional Array:
1 2 3
4 5 6

Two indices are required: One for the row and one for the column (matrix[row][column]).
Elements are stored in a contiguous memory block.
Nested loops are commonly used for traversal.
Useful for matrices, tables, and grid-based applications.

Example:

```
#include <stdio.h>
int main() {
    // Declare and initialize a 2D array
    int matrix[2][3] = {
        {1, 2, 3},
        {4, 5, 6}
    };
    // Print elements in matrix form
printf("Two-Dimensional Array (Matrix):\n");
    for (int i = 0; i< 2; i++) {
        for (int j = 0; j < 3; j++) {
printf("%d ", matrix[i][j]);
        }
printf("\n");
    }
    return 0;
}
```

Output:

Two-Dimensional Array (Matrix):
1 2 3
4 5 6

13.2.3. Multi-Dimensional Array (3D Array)

A multi-dimensional array extends the concept of 2D arrays to higher dimensions.
A multidimensional array in C is an array that contains multiple dimensions (more than one), such as 2D, 3D, or higher. It is used to represent complex data structures like matrices, 3D models, or multi-layered tables.

Syntax of Multidimensional Arrays:

data_typearray_name[size1][size2][size3]...[sizeN];
data_type → Type of elements stored in the array (e.g., int, float, char).
array_name → Name of the array.
size1, size2, size3, ..., sizeN → Sizes for each dimension.
Examples of Multidimensional Arrays
1. Three-Dimensional Array (3D Array)
A 3D array can be imagined as a stack of 2D matrices.
Declaration and Initialization:

```
int cube[2][2][2] = {
    {{1, 2}, {3, 4}},
    {{5, 6}, {7, 8}}
};
```

This represents a 2 x 2 x 2 cube-like structure:
Layer 0:
1 2
3 4

Layer 1:
5 6
7 8
Accessing Elements:

```
printf("%d", cube[1][0][1]);  // Output: 6 (Layer 1, Row 0, Column 1)
```

Modifying an Element:

```
cube[0][1][0] = 9;  // Changes the value from 3 to 9
```

Traversing a 3D Array Using Nested Loops:

```
#include <stdio.h>
int main() {
    int cube[2][2][2] = {
        {{1, 2}, {3, 4}},
        {{5, 6}, {7, 8}}
    };
    // Printing the 3D array
printf("Three-Dimensional Array:\n");
    for (int i = 0; i< 2; i++) { // Layer loop
printf("Layer %d:\n", i);
        for (int j = 0; j < 2; j++) { // Row loop
            for (int k = 0; k < 2; k++) { // Column loop
printf("%d ", cube[i][j][k]);
            }
printf("\n");
        }
printf("\n");
    }
    return 0;
}
```

Output:

```
Three-Dimensional Array:
Layer 0:
1 2
3 4
```

```
Layer 1:
5 6
7 8
```

2. Four-Dimensional Array (4D Array)

A 4D array extends the concept of 3D arrays, adding another layer of depth.

```
int arr4D[2][2][2][2] = {
   {{{1, 2}, {3, 4}}, {{5, 6}, {7, 8}}},
   {{{9, 10}, {11, 12}}, {{13, 14}, {15, 16}}}
};
```

Here, 4 indices are required to access elements: arr4D[block][layer][row][column].

Key Points:

Multidimensional arrays extend the concept of 1D and 2D arrays.
Each additional dimension increases complexity and memory usage.
Used in matrices, graphics, scientific computing, and simulations.
Access elements using multiple indices (array[i][j][k]...).

Multidimensional arrays

Example :

```
#include <stdio.h>
int main() {
   // Declare and initialize a 3D array
   int cube[2][2][2] = {
      {{1, 2}, {3, 4}},
      {{5, 6}, {7, 8}}
   };
   // Print elements
printf("Three-Dimensional Array:\n");
   for (int i = 0; i< 2; i++) {
      for (int j = 0; j < 2; j++) {
         for (int k = 0; k < 2; k++) {
printf("%d ", cube[i][j][k]);
         }
printf("\n");
      }
printf("\n");
   }
   return 0;
}
```

Output

```
Three-Dimensional Array:
1 2
3 4
5 6
7 8
```

13.3. CHARACTER ARRAY (STRING):

A character array is used to store and manipulate strings in C.

In C, a character array is used to store a sequence of characters, commonly known as a string. Since C does not have a built-in string data type, character arrays are used to store and

manipulate strings. A string in C is always terminated with a null character (\0), which indicates the end of the string.

Declaring and Initializing Character Arrays

1. Declaration of a Character Array:

```
char array_name[size];  // Declares a character array with a specific size
```

2. Initialization of a String:

```
char str1[] = "Hello";  // Automatically adds '\0' at the end
char str2[6] = {'H', 'e', 'l', 'l', 'o', '\0'};  // Manually adds '\0'
```

Input and Output of Strings in C

1. Using scanf() (Single Word Input)

```
char str[20];
scanf("%s", str);  // Reads a single word (stops at spaces)
```

2. Using gets() (Reads a Full Line - Deprecated)

```
char str[50];
gets(str);  // Reads a full line but is unsafe (deprecated)
```

3. Using fgets() (Safe Method for Multi-Word Input)

```
char str[50];
fgets(str, sizeof(str), stdin);  // Reads a full line safely
```

13.4. STRING FUNCTIONS IN C:

C provides several built-in string functions in the <string.h> library for common operations:

1. strlen(str) - Get String Length

```
strlen(str);  // Returns the number of characters in the string (excluding '\0')
```

2. strcpy(dest, src) - Copy String

```
strcpy(destination, source);  // Copies source string into destination
```

3. strcat(dest, src) - Concatenate Strings

```
strcat(destination, source);  // Appends source string to destination
```

4. strcmp(str1, str2) - Compare Strings

```
strcmp(str1, str2);  // Returns 0 if strings are equal, <0 or >0 otherwise
```

Key Points:

Character arrays store text (strings).

Strings are always terminated with a null character (\0).

Use fgets() instead of scanf() for multi-word input.

String functions like strlen(), strcpy(), strcat(), and strcmp() help in string operations.

13.5. ARRAY OF POINTERS IN C:

In C, an array of pointers is an array where each element is a pointer that stores the address of another variable. Instead of storing actual values, an array of pointers stores memory addresses, allowing efficient handling of data such as strings or dynamic memory allocation.

Declaring and Initializing an Array of Pointers

1. Declaration of an Array of Pointers:

```
data_type *array_name[size];
```

data_type specifies the type of data the pointers will point to.

size defines the number of pointers in the array.

2. Example: Array of Integer Pointers

```
#include <stdio.h>
```

```
int main() {
   int a = 10, b = 20, c = 30;
   int *arr[3];  // Array of 3 integer pointers
arr[0] = &a;  // Storing the address of a
arr[1] = &b;  // Storing the address of b
arr[2] = &c;  // Storing the address of c
printf("Values using array of pointers:\n");
   for (int i = 0; i< 3; i++) {
printf("%d ", *arr[i]);  // Dereferencing the pointers
   }
   return 0;
}
```

Output:

Values using array of pointers:
10 20 30

13.6. Array of Pointers to Strings (String Array):

An array of character pointers is commonly used to store multiple strings.
Example: Storing Strings in an Array of Pointers

```
#include <stdio.h>
int main() {
   char *str[] = {"Apple", "Banana", "Cherry", "Date"};

printf("Fruits List:\n");
   for (int i = 0; i< 4; i++) {
printf("%s\n", str[i]);
   }
   return 0;
}
```

Output:

Fruits List:
Apple
Banana
Cherry
Date

Advantages of Using an Array of Pointers

Efficient Memory Usage – Only memory for pointers and actual data is allocated, reducing memory wastage.Faster String Manipulation – Strings stored as pointers can be easily modified without copying.Dynamic Allocation – Arrays of pointers allow dynamic memory management using malloc(), calloc(), and free().

13.7. Dynamic Array (Using malloc):

A dynamic array in C is an array whose size is determined at runtime using dynamic memory allocation. The malloc() function from the <stdlib.h> library is used to allocate memory dynamically.
Syntax of malloc() for Dynamic Arrays

```
data_type *array_name = (data_type*) malloc(size * sizeof(data_type));
```

malloc(size) allocates size bytes of memory.
(data_type*) is used for type casting.
sizeof(data_type) ensures correct memory allocation.

Example: Dynamic Integer Array Using malloc()

```
#include <stdio.h>
#include <stdlib.h>
int main() {
   int n, *arr;
printf("Enter number of elements: ");
scanf("%d", &n);
   // Dynamic memory allocation
arr = (int*) malloc(n * sizeof(int));
   if (arr == NULL) {
printf("Memory allocation failed!");
      return 1;
   }
   // Storing values
printf("Enter %d integers:\n", n);
   for (int i = 0; i< n; i++) {
scanf("%d", &arr[i]);
   }
   // Printing values
printf("Entered elements: ");
   for (int i = 0; i< n; i++) {
printf("%d ", arr[i]);
   }
   // Free allocated memory
   free(arr);
   return 0;
}
```

Example Input & Output

Input:

Enter number of elements: 5
Enter 5 integers: 10 20 30 40 50

Output:

Entered elements: 10 20 30 40 50

13.8 SUMMARY TABLE FOR ARRAYS:

Array Type	Description	Example Use
One-Dimensional Array	A simple list of elements	`int arr[5] = {1, 2, 3, 4, 5};`
Two-Dimensional Array	An array of arrays (matrix)	`int matrix[2][3] = {{1,2,3},{4,5,6}};`
Multi-Dimensional Array	Arrays with more than 2 dimensions	`int cube[2][2][2] = {{{1,2},{3,4}},{{5,6},{7,8}}};`
Character Array (String)	Used to store text data	`char str[] = "Hello";`
Array of Pointers	Stores addresses of other variables	`char *names[] = {"Alice", "Bob"};`
Dynamic Array	Allocated at runtime	`int *arr = (int*)malloc(5 * sizeof(int));`

13.9 SUMMARY:

An array in C is a data structure that stores multiple values of the same type in contiguous memory locations. Arrays help in organizing data efficiently by allowing indexed access to elements. They can be one-dimensional (1D), two-dimensional (2D), or multi-dimensional (3D, etc.).Arrays must be declared before use, specifying data type, name, and size. Initialization can be done at declaration or later using loops. Elements are accessed using indices, with indexing starting from 0.Traversal is done using loops like for, while, or do-while. Arrays are useful for mathematical computations, data storage, and matrix operations. C supports different types of arrays, including numeric, character (strings), and multi-dimensional arrays. Understanding arrays is crucial for efficient data handling and algorithm implementation in C.

13.10 KEY TERMS:

Array, Declaration, Initialization, Indexing, Traversal, Data Type, Contiguous Memory, One-Dimensional, Two-Dimensional, Multi-Dimensional.

13.11 REVIEW QUESTIONS:

1. What is an array in C, and how does it help in managing multiple values?
2. Explain the syntax for declaring a one-dimensional array in C?
3. How can an array be initialized without explicitly specifying its size? Provide an example.
4. What is the difference between a one-dimensional and a two-dimensional array?
5. How do you access and modify elements in an array using indexing? Give an example.
6. What is the purpose of using loops for array traversal? Provide a simple C program demonstrating traversal.
7. Explain the significance of the null character (‘\0’) in character arrays (strings) in C?

13.12 SUGGESTED READINGS:

1. Programming Using the C Language: Hutchison, Robert C Mac Graw Hill
2. C Programming, E Balaguruswamy, Tata MC Graw Hill.
3. Let us C, Yaswanth, P.Kanetkar, BPB Publishers.

Dr. B.Basaveswara Rao

LESSON-14

CYBERSECURITY: THREATS, ATTACKS, AND MALICIOUS SOFTWARE

OBJECTIVES:

The objectives of the lesson are

1. Gain in-depth knowledge of various cybersecurity threats, attack techniques, and vulnerabilities that affect digital systems, including phishing, ransomware, DoS attacks, and malware infections.
2. Explore how cybercriminals exploit security weaknesses and learn effective defense mechanisms, including threat detection, risk assessment, and implementation of cybersecurity best practices.
3. Enhance skills in identifying, preventing, and mitigating cyber threats by applying advanced security solutions, malware defense techniques, and cybersecurity frameworks to safeguard systems and data.

STRUCTURE:

14.1 Introduction to Cyber Security
- **14.1.1 What is Cyber Security**
- **14.1.2. Importance of Cyber Security**
- **14.1.3. Cyber Security Fundamentals**

14.2. Security and Ethical Challenges in IT in Cybersecurity
- **14.2.1. Introduction**
- **14.2.2. Security Challenges in IT**
- **14.2.3. Ethical Challenges in IT and Cybersecurity**
- **14.2.4. Best Practices to Address Security and Ethical Challenges**

14.3. Need for Security in Cybersecurity
- **14.3.1. Introduction**
- **14.3.2. What is Security in Cybersecurity**
- **14.3.2. What is Security in Cybersecurity**
- **14.3.3. Why is Security Needed in Cybersecurity**
- **14.3.4. Real-Life Examples of Security Failures**

14.4. Security Threats and Attacks
- **14.4.1. Types of Cyber Attacks**
- **14.4.2. Vulnerabilities, Threats, and Harmful Acts**
- **14.4.3. Security Vulnerabilities, Threats, and Attacks**
- **14.4.3. The 7 Layers of Cyber Security**
- **14.4.5. Computer Criminals**
- **14.4.6. Assets and Threats**

14.5. Malicious Software (Malware)
- **14.5.1. What is Malware**
- **14.5.2. Introduction**
- **14.5.3. Types of Malware**
- **14.5.4. How Malware Spreads**
- **14.5.5. Signs of Malware Infection**

14.5.6. Preventing Malware Attacks
14.5.7. Malware Detection and Removal

14.6. Summary

14.7. Key Terms

14.8. Review Questions

14.9. Suggested Readings

14.1 INTRODUCTION TO CYBER SECURITY:

Cyber security has become a critical concern due to the rapid increase in cyber threats and attacks. Attackers are employing increasingly sophisticated techniques to compromise systems, affecting individuals, small businesses, and large organizations alike. As a result, both IT and non-IT firms recognize the importance of cyber security and are adopting various measures to mitigate cyber threats effectively.

14.1.1 What is Cyber Security

Cyber security involves the protection of systems, networks, and data from cyber threats, unauthorized access, and attacks. It encompasses a combination of people, processes, and technology working together to reduce vulnerabilities, deter threats, and ensure data integrity, confidentiality, and availability.

Cyber security can be defined as:

- The practice of protecting networks, computers, programs, and data from attacks, damage, or unauthorized access.
- A set of techniques and practices designed to secure digital data, whether stored, transmitted, or in use.
- The protection of internet-connected systems, including hardware, software, and data, from cyber threats.

14.1.2. Importance of Cyber Security

In today's digital world, cyber security plays a vital role for several reasons:

- **Financial Impact:** Cyber attacks can lead to significant financial losses for businesses.
- **Reputational Damage:** A data breach can severely harm an organization's reputation and customer trust.
- **Increasing Threats:** Cybercriminals continue to develop more advanced attack techniques.
- **Regulatory Compliance:** Regulations such as GDPR require organizations to safeguard personal data.

Due to these reasons, cyber security is now an essential part of business operations. Organizations are focusing on developing response plans to minimize the impact of cyber attacks. However, effective response planning requires a solid understanding of cyber security fundamentals.

14.1.3. Cyber Security Fundamentals

1. Confidentiality

Confidentiality ensures that sensitive information is not accessed by unauthorized individuals. It also involves protecting the identity of authorized users handling the data.

Breaches in confidentiality often occur due to weak encryption, man-in-the-middle (MITM) attacks, and accidental data exposure.

Measures to ensure confidentiality:

- Data encryption
- Two-factor authentication
- Biometric verification
- Security tokens

2. Integrity

Integrity focuses on maintaining the accuracy and reliability of data by preventing unauthorized modifications.

Measures to ensure integrity:

- Cryptographic checksums
- File permissions
- Uninterrupted power supplies
- Regular data backups

3. Availability

Availability ensures that authorized users can access the necessary information when required. Cyber attacks like Distributed Denial of Service (DDoS) can compromise availability.

Measures to ensure availability:

- Data backups on external drives
- Implementation of firewalls
- Backup power supplies
- Data redundancy

By understanding and implementing these core principles, individuals and organizations can enhance their cyber security posture and protect themselves against cyber threats effectively.

14.2. SECURITY AND ETHICAL CHALLENGES IN IT IN CYBERSECURITY:

14.2.1. Introduction

With the rise of digital transformation and increasing reliance on information technology (IT), cybersecurity has become a cornerstone of modern business and communication. However, along with technological advancements, several security and ethical challenges have emerged, posing risks to individuals, organizations, and nations.

14.2.2. Security Challenges in IT

Data Breaches:

- Unauthorized access to sensitive information such as personal data, financial records, or intellectual property.
- Examples: Credit card leaks, health record thefts.

Malware and Ransomware Attacks:

- Malicious software that damages, disrupts, or gains unauthorized access to systems.
- Ransomware encrypts files and demands payment for decryption.

Phishing and Social Engineering:

- Fraudulent attempts to obtain sensitive information by pretending to be trustworthy entities.
- Increasingly sophisticated tactics, including spear-phishing.

Insider Threats:
- Employees or contractors who misuse their access to cause harm intentionally or unintentionally.
- Hard to detect and prevent.

Denial-of-Service (DoS) Attacks:
- Attacks that flood systems with traffic to make services unavailable.
- Distributed DoS (DDoS) attacks are more complex and widespread.

Cloud Security Risks:
- Data stored in cloud services can be vulnerable if not properly encrypted or monitored.
- Misconfigured cloud environments often lead to data exposure.

Internet of Things (IoT) Vulnerabilities:
- Many IoT devices lack proper security, making them easy targets for cybercriminals.
- Can be exploited for botnets or surveillance.

14.2.3. Ethical Challenges in IT and Cybersecurity

Privacy Concerns:
- How organizations collect, use, store, and share personal data.
- Ethical issues around consent, transparency, and surveillance.

Ethical Hacking:
- White-hat hackers test systems for vulnerabilities, but ethical boundaries can be blurred.
- Is it ethical to hack a system to prove a point?

Use of Artificial Intelligence:
- AI in cybersecurity raises questions about bias, autonomy, and decision-making.
- Risk of AI being used for malicious purposes like deepfakes or automated hacking.

Digital Rights and Freedom:
- Balancing cybersecurity with freedom of expression and access to information.
- Ethical dilemmas in censorship and internet regulation.

Responsibility and Accountability:
- Who is responsible when a security breach occurs: developers, users, or organizations?
- Lack of clear accountability frameworks.

Whistle blowing:
- Ethical challenges faced by individuals exposing unethical or illegal IT practices.
- Risk of retaliation and legal implications.

14.2.4. Best Practices to Address Security and Ethical Challenges

- Implement strong security policies and access controls.
- Conduct regular cybersecurity training and awareness programs.
- Ensure data encryption and multi-factor authentication.
- Develop an ethical code of conduct for IT professionals.
- Conduct ethical reviews and audits of technology use.
- Promote transparency and user consent in data collection.
- Adopt zero-trust security models and continuous monitoring.

Security and ethical challenges in IT and cybersecurity are interconnected and evolving. Addressing them requires a proactive approach involving technology, policy, and a strong ethical framework. As the digital world grows, so does our responsibility to protect it while upholding integrity, privacy, and trust.

14.3. NEED FOR SECURITY IN CYBERSECURITY:

14.3.1. Introduction

In today's digital world, vast amounts of sensitive data are stored, processed, and transmitted through computer systems and networks. The increasing dependence on technology has also increased the potential for cyber threats. This makes security not just an option but a necessity in cybersecurity.

14.3.2. What is Security in Cybersecurity

Security in cybersecurity refers to the measures, practices, and technologies used to protect systems, networks, programs, and data from unauthorized access, attacks, or damage.

14.3.3. Why is Security Needed in Cybersecurity

Protection of Confidential Data:

- Sensitive information like personal, financial, or business data must be kept secure.
- Data breaches can result in identity theft, financial loss, or legal consequences.

Preventing Unauthorized Access:

- Security mechanisms ensure only authorized users can access systems or data.
- Protects against hackers, insider threats, and unauthorized users.

Maintaining Data Integrity:

- Ensures that data is accurate, consistent, and unaltered during storage or transmission.
- Crucial for banking, healthcare, government, and other sensitive sectors.

Ensuring Availability of Services:

- Security helps prevent service interruptions caused by cyberattacks (like DDoS).
- Ensures continuous operation of critical systems (e.g., hospitals, airlines, banks).

Compliance with Legal and Regulatory Standards:

- Many industries are required by law to implement strong cybersecurity measures (e.g., GDPR, HIPAA).
- Non-compliance can lead to heavy penalties and reputational damage.

Protecting Business Reputation and Trust:

- A single cyberattack can damage an organization's brand and customer trust.
- Security builds customer confidence and business credibility.

Countering the Rise in Cybercrime:

- Cyberattacks are increasing in frequency and sophistication (ransomware, phishing, etc.).
- Robust security is needed to stay ahead of evolving threats.

Supporting National Security and Critical Infrastructure:

- Cybersecurity protects national infrastructure like power grids, transport systems, and communication networks.
- Prevents cyber warfare and state-sponsored attacks.

14.3.4. Real-Life Examples of Security Failures

- Yahoo (2013-14): Over 3 billion accounts compromised due to poor encryption.
- Equifax (2017): Data breach of 147 million records due to unpatched software vulnerability.
- WannaCry Ransomware (2017): Affected systems worldwide due to outdated Windows OS.

The need for security in cybersecurity is more critical than ever. It is essential to protect data, systems, and infrastructure from ever-growing cyber threats. A secure cyber environment ensures business continuity, legal compliance, and public trust in technology.

14.4. SECURITY THREATS AND ATTACKS:

14.4.1. Types of Cyber Attacks

A cyber attack is an exploitation of computer systems and networks. It involves the use of malicious code to alter computer logic, data, or code, leading to cybercrimes such as information and identity theft.

Cyber attacks can be broadly categorized into two types:

1. Web-based attacks
2. System-based attacks

1.Web-based Attacks:

These attacks target websites or web applications. Some of the most common web-based attacks include:

1. **Injection Attacks**
   These attacks involve injecting malicious data into a web application to manipulate it and extract sensitive information. *Examples:* SQL Injection, Code Injection, Log Injection, XML Injection.
2. **DNS Spoofing**
   A hacking technique that injects false data into a DNS resolver's cache, causing it to return incorrect IP addresses. This redirects traffic to an attacker's system, potentially leading to significant security risks.
3. **Session Hijacking**
   A security attack that targets user sessions over protected networks. Attackers steal cookies to gain unauthorized access to user data.
4. **Phishing**
   A deceptive attack where cybercriminals masquerade as trustworthy entities to steal sensitive information like login credentials and credit card details.
5. **Brute Force Attack**
   This attack uses trial and error to guess passwords or PINs by generating numerous guesses until the correct one is found.
6. **Denial of Service (DoS) Attack**
   This attack aims to make a server or network resource unavailable by flooding it with traffic or sending data that causes a crash.
   a. *Volume-based attacks:* Saturate bandwidth (measured in bits per second).
   b. *Protocol attacks:* Consume server resources (measured in packets per second).
   c. *Application layer attacks:* Overload the web server (measured in requests per second).
7. **Dictionary Attack**
   This attack involves using a pre-stored list of commonly used passwords to crack user accounts.
8. **URL Interpretation Attack**
   This attack manipulates specific parts of a URL to access unauthorized web pages.
9. **File Inclusion Attacks**
   Attackers exploit vulnerabilities to access or execute unauthorized files on a web server.
10. **Man-in-the-Middle (MITM) Attack**
    An attacker intercepts the communication between a client and a server, allowing them to read, modify, or inject data into the intercepted connection.

2.System-based Attacks:
These attacks aim to compromise computers or networks. Some of the most common system-based attacks include:
1. **Virus**
A self-replicating malicious program that spreads through computer files without user knowledge, often causing damage to the system.
2. **Worm**
Similar to a virus, a worm spreads across networks, often via email attachments, infecting unprotected systems.
3. **Trojan Horse**
A deceptive program that appears legitimate but executes malicious actions when run, altering system settings and behavior.
4. **Backdoors**
A hidden method of bypassing normal authentication to gain unauthorized access to an application or operating system.
5. **Bots**
Automated programs that interact with network services. While some are used for legitimate purposes, malicious bots can perform cyber attacks such as data theft and denial-of-service attacks.
Understanding these cyber attacks and their preventive measures is crucial for strengthening security and mitigating threats.

14.4.2. Vulnerabilities, Threats, and Harmful Acts

No system is completely immune to cyber threats. Organizations handling data must implement monitoring mechanisms to identify vulnerabilities and mitigate security risks promptly.

Understanding Cyber Threats and Vulnerabilities

- Cyber Threats – Security incidents that pose risks to data systems. Examples include phishing attacks that install malware, insider errors leading to breaches, or natural disasters disrupting data access.
- Vulnerabilities – Weaknesses in a system that threat actors exploit. Examples include SQL injections, server misconfigurations, cross-site scripting, and unencrypted data transmission.

The risk level of a cyber threat is determined by multiplying its probability by the potential loss it could cause.

14.4.3. Security Vulnerabilities, Threats, and Attacks

Categories of Vulnerabilities:

- Corrupted (Loss of Integrity) – Unauthorized changes to data.
- Leaky (Loss of Confidentiality) – Exposure of sensitive information.
- Unavailable or Slow (Loss of Availability) – Disruptions affecting data accessibility.

Threats vs. Attacks:

- Threats – Potential risks to assets due to vulnerabilities.
- Attacks – Realized threats that impact security.
    - Passive Attacks – Extract information without affecting system resources.
    - Active Attacks – Alter system resources or disrupt operations.
    - Insider Attacks– Initiated by internal employees or partners.
    - Outsider Attacks– Launched by external cybercriminals.

14.4.4. The 7 Layers of Cyber Security

Effective cyber security revolves around protecting mission-critical assets. The seven layers of cyber security include:

1. **Mission Critical Assets** – The data and resources that need protection.
2. **Data Security** – Controls that secure data storage and transmission.
3. **Application Security** – Measures that protect access to applications and the sensitive data they handle.
4. **Endpoint Security** – Safeguards for device connections to the network.
5. **Network Security** – Controls that prevent unauthorized access and ensure network integrity.
6. **Perimeter Security** – A combination of physical and digital security measures to protect the business as a whole.
7. **The Human Layer** – Employees are often the weakest link in cyber security. Human security controls include phishing simulations and access management to mitigate risks posed by cybercriminals, malicious insiders, and negligent users.

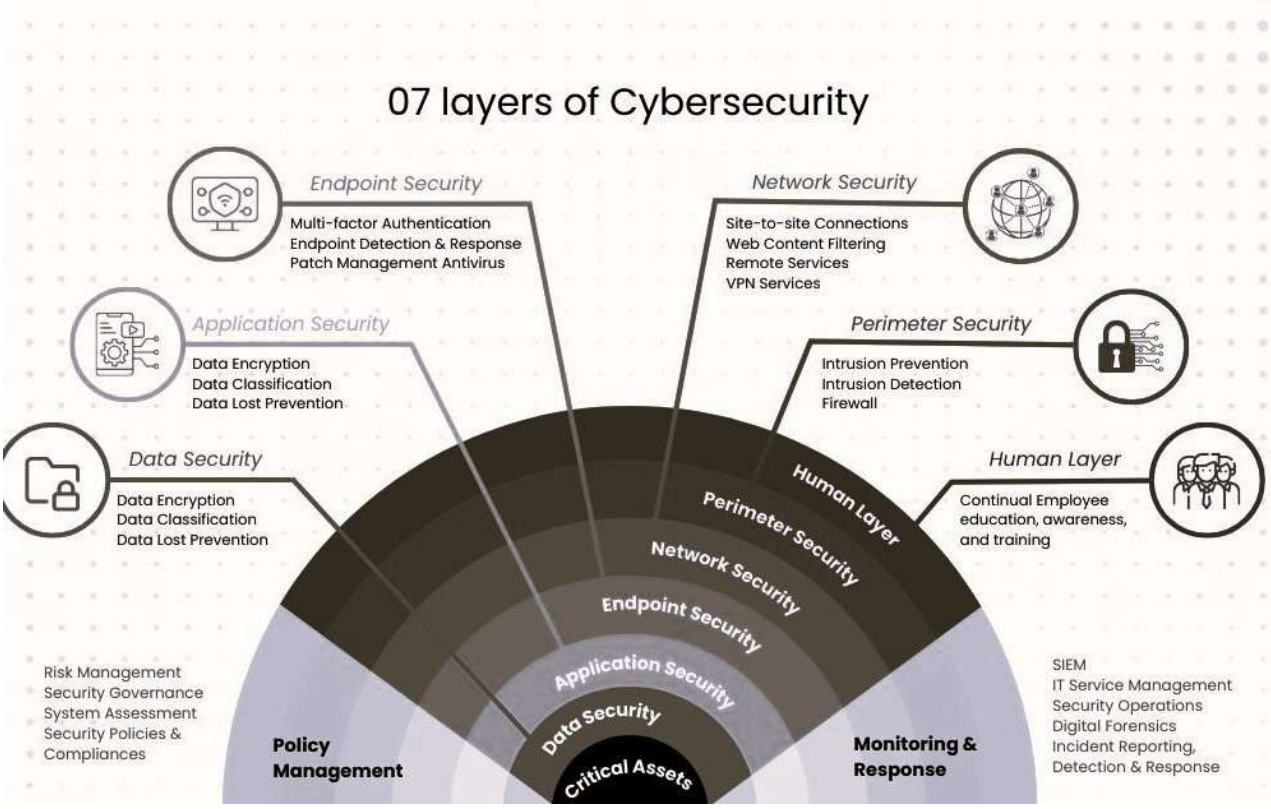


Figure.14.1 The 7 Layers of Cyber Security.

14.4.5. Computer Criminals

Computer criminals leverage software, hardware, and data to conduct cybercrimes, threatening businesses and government institutions worldwide. Cybercrime involves any illegal activity aided by a computer. Understanding the motivations and behaviors of cybercriminals helps in developing preventive measures to mitigate security threats.

The CIA Triad

The CIA Triad is a foundational security model that focuses on three key aspects of IT security:

1. Confidentiality
- Protects sensitive information from unauthorized access.
- Implemented using access control lists, encryption, and permissions.

2. Integrity
- Ensures data is not modified or deleted by unauthorized parties.
- Implemented through cryptographic hashes, version control, and backup systems.

3. Availability
- Ensures authorized users can access necessary data and systems.
- Implemented through redundancy, failover mechanisms, and secure authentication.

While the CIA Triad is essential for information security, organizations must also consider other security concerns, such as resource misuse and unauthorized access to hardware.

14.4.6. Assets and Threats

What is an Asset

An asset includes any data, device, or system component valuable to an organization. Assets are often valuable due to the sensitive data they contain or the access they provide.

Examples of Assets:
- Employee devices (laptops, desktops, company phones).
- Applications and software platforms.
- Critical infrastructure (servers, databases, network systems).
- Physical and digital information storage systems.

Understanding and categorizing assets is crucial for organizations to implement effective security measures and mitigate cyber threats.

14.5. MALICIOUS SOFTWARE (MALWARE):

14.5.1. What is Malware

Malware, short for malicious software, is any program or code designed to harm computer systems, networks, or users. It can infiltrate devices without user consent, stealing sensitive data like bank details and passwords, displaying intrusive advertisements, or altering system settings. Malware commonly spreads through:
- Free software downloads
- Clicking on suspicious links
- Opening emails from unknown sources
- Visiting unsafe websites
- Lack of updated antivirus software

14.5.2. Introduction

Malicious software, commonly known as malware, is a type of software designed to disrupt, damage, or gain unauthorized access to computer systems and networks. Cybercriminals use malware to steal sensitive information, damage files, or control a system without the user's knowledge. With the increasing reliance on digital technology, malware threats have become a significant concern for individuals, businesses, and governments.

14.5.3. Types of Malware

Malware comes in various forms, each serving a specific malicious purpose. The most common types include:

1. **Viruses**: Self-replicating programs that attach to files and spread when executed.

Symptoms:

- Falling text effects on the screen
- Slow system performance
- Reduced free memory
- Hard disk running out of space
- Boot failures

Common Virus Types:

- **Parasitic:** Attaches to executable files (.COM, .EXE), e.g., Jerusalem.
- **Boot Sector:** Infects boot drives, e.g., Polyboot, Disk Killer.
- **Polymorphic:** Alters itself to avoid detection, e.g., Cascade.
- **Memory Resident:** Resides in memory, damaging active files, e.g., Randex.
- **Stealth:** Modifies itself to avoid detection, e.g., Frodo.
- **Macro:** Spreads through infected documents, e.g., Melissa.
- **Hybrid:** Combines multiple virus traits, e.g., Happy99 (Email virus).

2. Worms: Standalone malware that replicates itself across networks without user intervention.

Types of Worms:

- **Email Worm:** Spreads via infected email attachments.
- **Instant Messaging Worm:** Exploits messaging apps.
- **Internet Worm:** Targets OS vulnerabilities.
- **IRC Worm:** Distributes infected files.
- **Payload Worm:** Deletes files or installs backdoors.
- **Good Intent Worm:** Installs updates.

3. Trojans: Malicious software disguised as legitimate applications to trick users into execution.

4. Ransomware: Encrypts files and demands payment for their release.

5. Spyware: Secretly gathers user information and transmits it to attackers.

6. Adware: Displays intrusive advertisements and may collect user data.

7. Rootkits: Conceals malicious activities and grants attackers privileged access.

Types:

- **Application Rootkits:** Replace system files.
- **Kernel Rootkits:** Target the operating system core.
- **Methods of Infection:** DLL injection, kernel object manipulation, and hooking.

8. Keyloggers: Records keystrokes to steal credentials and sensitive data.

9. Botnets: Networks of infected computers controlled remotely for malicious purposes.

14.5.4. How Malware Spreads

Malware spreads through various attack vectors, including:

- **Email Attachments**: Phishing emails containing infected files or links.
- **Malicious Websites**: Drive-by downloads that automatically install malware.
- **Software Downloads**: Free or pirated software bundled with malware.
- **USB Devices**: Infected removable media transferring malware to a system.

- **Social Engineering**: Manipulating users into downloading and executing malicious programs.

14.5.5. Signs of Malware Infection

A system infected with malware may exhibit the following symptoms:

- Slow system performance
- Frequent crashes or freezes
- Unexpected pop-up ads
- Unauthorized changes to system settings
- Unusual network activity
- Disabled security programs

14.5.6. Preventing Malware Attacks

To mitigate malware threats, consider the following preventive measures:

- Install and regularly update antivirus and anti-malware software.
- Keep operating systems and applications up to date.
- Avoid clicking on suspicious links or downloading files from unknown sources.
- Use strong, unique passwords and enable multi-factor authentication.
- Implement network security measures like firewalls and intrusion detection systems.
- Regularly back up important files to secure locations.

14.5.7. Malware Detection and Removal

If a system is infected, follow these steps to detect and remove malware:

1. Run a full system scan using updated antivirus software.
2. Disconnect the affected device from the network.
3. Boot into Safe Mode and remove suspicious programs.
4. Restore the system from a clean backup if necessary.
5. Reset passwords and monitor accounts for unusual activity.

Advantages of Malware Detection and Removal:

- **Enhanced Security:** Protects systems from cyber threats.
- **Prevention of Data Loss:** Safeguards sensitive files.
- **Reputation Protection:** Prevents damage to organizational credibility.
- **Increased Productivity:** Eliminates performance slowdowns.

Disadvantages of Malware Detection and Removal:

- **Time-Consuming:** Requires specialized expertise.
- **Costs:** Premium security solutions can be expensive.
- **False Positives:** Some tools may mistakenly flag safe programs.
- **Complexity:** Constantly evolving threats make detection challenging.
- **Risk of Data Loss:** Some removal tools can inadvertently delete important files.

Note: Malware poses a significant threat to cybersecurity, affecting individuals, businesses, and organizations. Awareness and proactive security measures are essential to prevent infections and minimize damage. By staying informed and adopting best security practices, users can safeguard their systems and data against malware threats.

14.6. SUMMARY:

Cyber security is essential in today's digital world due to the rise in cyber threats affecting individuals and organizations. It involves protecting systems, networks, and data from unauthorized access and attacks through encryption, authentication, and security protocols.

The core principles include confidentiality, integrity, and availability, ensuring data remains secure and accessible. Cyber attacks, such as phishing, malware, and denial-of-service attacks, exploit system vulnerabilities, causing financial and reputational damage.

Organizations implement layered security measures, including network, application, and endpoint security, to mitigate risks. The CIA triad—confidentiality, integrity, and availability—forms the foundation of cyber security. Malware, including viruses, ransomware, and spyware, spreads through phishing emails, malicious websites, and social engineering tactics. Preventive measures like antivirus software, firewalls, and multi-factor authentication help reduce cyber risks. Organizations also enforce regulatory compliance and security awareness training to strengthen cyber defense. Understanding cyber security fundamentals is crucial in combating cyber threats and ensuring a secure digital environment.

14.7. KEY TERMS:

Cyber Security, Confidentiality, Integrity, Availability, Cyber Attacks, Malware, Phishing, Encryption, Firewall, Multi-Factor Authentication.

14.8. REVIEW QUESTIONS:

1. What is cyber security, and what are its key objectives?
2. Explain the different types of cyber attacks with examples?
3. What are the core principles of cyber security, and how do they help in protecting data?
4. Differentiate between viruses, worms, and trojans in terms of their behavior and impact?
5. What are some effective measures to prevent malware infections and cyber threats?

14.9. SUGGESTED READINGS:

1. "Cybersecurity: The Essential Body of Knowledge" – by Dan Shoemaker
2. "Principles of Information Security" – by Michael E. Whitman & Herbert J. Mattord
3. "Computer Security: Principles and Practice" – by William Stallings & Lawrie Brown
4. "Network Security Essentials: Applications and Standards" – by William Stallings
5. "The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" – by Dafydd Stuttard& Marcus Pinto
6. "Hacking: The Art of Exploitation" – by Jon Erickson
7. "Applied Cryptography: Protocols, Algorithms, and Source Code in C" – by Bruce Schneier

Dr. B. Basaveswara Rao

LESSON- 15

HACKING, CRYPTOGRAPHY, AND DIGITAL SIGNATURES

OBJECTIVES:

The objectives of the lesson are

1. To understand the fundamentals of hacking, including ethical hacking practices and cybersecurity measures to protect digital systems.
2. To explore cryptography techniques for securing data transmission, ensuring confidentiality, integrity, and authentication.
3. To analyze the role of digital signatures in verifying authenticity, preventing forgery, and enhancing secure communication.
4. To apply advanced security strategies to mitigate cyber threats and strengthen digital infrastructure.

STRUCTURE:

15.1. Hacking

- **15.1.1. Introduction to Hacking in Cybersecurity**
- **15.1.2. History of Hacking**
- **15.1.3. Types of Hacking in Cybersecurity**
- **15.1.3.1. White Hat Hackers: The Ethical Protectors**
- **15.1.3.2. Black Hat Hackers: The Cybercriminals**
- **15.1.3.3. Grey Hat Hackers: The Middle Ground**
- **15.1.4. Hacking Security Mechanisms in Cybersecurity**
- **15.1.5. Types of Hacking Security Mechanisms**

15.2. Cryptography

- **15.2.1. Key Principles of Cryptography**
- **15.2.2. Types of Cryptography**
- **15.2.3. Applications of Cryptography in Cybersecurity**
- **15.3. Digital Signatures**
- **15.3.1. How Digital Signatures Work**
- **15.3.2. Process of Digital Signing and Verification**
- **15.3.3. Examples of Digital Signatures in Cybersecurity**

15.4 Summary

15.5 Key Terms

15.6 Review Questions

15.7 Suggested Readings

15.1 HACKING:

Hacking (also called cyber hacking), involves using unauthorized or unconventional methods to access digital devices, computer systems, or networks. A typical example is a cybercriminal exploiting security flaws to breach a network and steal sensitive data.

15.1.1 Introduction to Hacking in Cybersecurity

Hacking in cybersecurity refers to the practice of gaining access to computer systems, networks, or data—either ethically or maliciously—by exploiting vulnerabilities. While hacking is often associated with cybercriminals who steal sensitive information or disrupt operations, it also plays a vital role in strengthening security. Ethical hackers, also known as white hat hackers, help organizations identify and fix weaknesses before malicious hackers can exploit them. As technology evolves, hacking techniques continue to advance, making cybersecurity a crucial aspect of digital safety for individuals, businesses, and governments worldwide.

15.1.2. History of Hacking

1960s : The Birth of Hacking

Hacking originated in 1961 at MIT, where students modified electronic model trains before shifting their focus to computers, experimenting with early IBM machines.

1970s–1980s: The Rise of Hacking Culture

With the launch of ARPANET in 1969, hackers became interconnected, forming a subculture. The 1970s saw the rise of "phreakers" who exploited telephone systems—Apple founders Steve Jobs and Steve Wozniak were among them. The 1980s brought personal computers, expanding hacking beyond universities and leading to increased cybercrime, including piracy and malware distribution.

1986–Present: Cyber security and Modern Hacking

The U.S. passed the Computer Fraud and Abuse Act in 1986 to combat malicious hacking. As the internet expanded in the 1990s and beyond, hacking evolved, targeting connected devices from refrigerators to cars. The late 1990s saw the emergence of computer worms and large-scale DDoS attacks, escalating cybersecurity threats.

Figure.15.1 Cyber Hacking.

15.1.3. Types of Hacking in Cybersecurity

Hacking in Shades of Black, White, and Grey

The classification of hackers into black, white, and grey hats is often linked to American Western films, where black hats symbolized villains and white hats represented heroes. However, in reality, these color associations weren't always consistent in movies.

In cybersecurity, **white hat hackers** are ethical professionals who protect systems from cyber threats, while **black hat hackers** exploit vulnerabilities for personal gain. **Grey hat hackers** fall in between—though their methods may be unauthorized, their intentions are often to expose security flaws for the greater good.

15.1.3.1. White Hat Hackers: The Ethical Protectors

White hat hackers work for organizations or governments, hacking systems with permission to identify and fix security vulnerabilities.

Ethical Hacking (White-Hat Hacking)

- Ethical hackers, also known as penetration testers, are cybersecurity professionals who legally test systems for vulnerabilities.

Example: A company hires a Certified Ethical Hacker (CEH) to conduct penetration testing and identify weak security points in its online banking system.

15.1.3.2. Black Hat Hackers: The Cybercriminals

Black hat hackers operate illegally, breaking into systems without consent, often for financial gain or to support other criminal activities.

Black-Hat Hacking (Malicious Hacking)

- Black-hat hackers engage in illegal activities such as stealing sensitive data, deploying malware, or disrupting services for financial gain or personal motives.

Example: A hacker launches a ransomware attack on a hospital's database, encrypting patient records and demanding a ransom for decryption.

15.1.3.3. Grey Hat Hackers: The Middle Ground

Grey hat hackers may access systems without authorization but usually do so with good intentions, such as exposing security risks to the public. However, despite their motives, their actions can still be considered illegal.

Gray-Hat Hacking (Unauthorized but Non-Malicious Hacking)

- Gray-hat hackers find security flaws without permission and may report them or request compensation. Their actions are not always legal but are not necessarily harmful.

Example: A security researcher discovers a vulnerability in a government website and notifies the authorities without causing harm.

Figure.15.2 Three types of Hackers.

15.1.4. Hacking Security Mechanisms in Cybersecurity

Hacking Security

Hacking Security refers to the practices and technologies used to protect computer systems, networks, and data from unauthorized access, damage, or theft caused by hackers. It plays a vital role in cybersecurity, helping to defend against both internal and external threats.

Objectives of Hacking Security:

- **Prevent** unauthorized access to systems and data
- **Detect** malicious activities or breaches
- **Respond** quickly to minimize damage
- **Recover** from cyberattacks efficiently

Key Components of Hacking Security:

1. **Authentication and Access Control**
   - Multi-Factor Authentication (MFA)
   - Role-Based Access Control (RBAC)
2. **Data Protection and Encryption**
   - End-to-End Encryption (SSL/TLS)
   - Data Masking to hide sensitive info
3. **Network Security**
   - Firewalls to filter traffic
   - Intrusion Detection and Prevention Systems (IDPS)
4. **Malware Protection**
   - Antivirus and anti-malware tools
   - Sandboxing to isolate and test suspicious files
5. **Security Awareness Training**
   - Educating users on phishing, scams, and safe practices
   - Regular workshops and simulated attacks
6. **Regular Security Audits and Ethical Hacking**
   - Penetration testing by white-hat hackers
   - Identifying and patching system vulnerabilities

Note: Hacking Security is essential in today's digital world. Organizations must stay proactive by adopting strong security measures and leveraging ethical hacking to guard against evolving cyber threats.

15.1.5. Types of Hacking Security Mechanisms

1. Phishing Attacks

Definition:

Phishing is a social engineering attack where cybercriminals trick individuals into revealing confidential information like passwords, credit card numbers, or personal data by posing as a legitimate source. (Or) Cybercriminals send fake emails or messages to trick users into revealing sensitive information like passwords or credit card details.

How It Works:

- Attackers send deceptive emails, messages, or fake websites that appear to come from trusted institutions (like banks, social media platforms, or employers).
- These messages often create a sense of urgency, asking the victim to "verify your account," "reset your password," or "confirm a transaction."
- Victims are then directed to a fraudulent website that looks real but is designed to steal their information.

Example:

A user receives an email that looks like it's from their bank, asking them to log in to resolve a suspicious transaction. The link in the email leads to a fake login page where the user's credentials are captured by the hacker.

Prevention Tips:

- Do not click on suspicious links or attachments in emails or messages.

- Always verify the sender's email address.
- Look for HTTPS in the website URL before entering sensitive data.
- Use multi-factor authentication (MFA) to add an extra layer of security.
- Keep antivirus software updated to detect phishing attempts.

2. Malware Attacks

Definition:

Malware (short for malicious software) refers to any software intentionally designed to damage, disrupt, or gain unauthorized access to a computer system, network, or device.(or) Hackers use malicious software (viruses, ransomware, spyware) to infect systems and steal or destroy data.

Types of Malware:

1. Viruses – Attach themselves to legitimate programs and spread when the host program runs.
2. Worms – Self-replicate and spread without human interaction.
3. Trojan Horses – Disguise as legitimate software but execute harmful functions.
4. Ransomware – Locks or encrypts data and demands payment for release.
5. Spyware – Secretly gathers user data and sends it to the attacker.
6. Adware – Automatically displays or downloads ads, sometimes with malicious intent.

Example:

The WannaCry ransomware attack in 2017 affected thousands of computers worldwide, locking users out of their systems and demanding payment in Bitcoin.

Prevention Tips:

- Install and regularly update antivirus and anti-malware software.
- Avoid downloading files or software from untrusted sources.
- Don't click on suspicious email attachments or pop-up ads.
- Keep your operating system and software updated to patch vulnerabilities.
- Use firewalls and email filters to block potential threats.

3. SQL Injection

Definition:

SQL Injection is a hacking technique where attackers inject malicious SQL commands into a web application's input fields (like login forms or search boxes) to access or manipulate the database.(or) Attackers exploit vulnerabilities in a website's database to gain unauthorized access and manipulate data.

How it Works:

When a web application fails to properly validate or sanitize user inputs, attackers can insert specially crafted SQL queries to:

- Bypass login authentication
- View, modify, or delete sensitive data
- Take control of the database

Example 1:

A hacker enters ' OR '1'='1 in the username field of a login form. If not handled properly, the query may return true, granting unauthorized access.

Example 2: A hacker inserts malicious SQL code into a website's login page to retrieve customer records from the database.

Impact:

- Data breaches
- Unauthorized account access
- Data corruption or deletion

- Loss of customer trust

Prevention:

- Use parameterized queries (prepared statements)
- Implement strong input validation
- Limit database privileges
- Regular security audits and code reviews

4. Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) Attacks

Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) Attacks

Definition:

DoS and DDoS attacks are cyberattacks aimed at making a system, service, or network unavailable to its intended users by overwhelming it with traffic or requests. (Or) Attackers overload a website or server with excessive traffic, making it inaccessible to users.

Types:

1. DoS Attack (Denial-of-Service):

- A single attacker uses one system to flood the target with traffic or requests.
- Example: Sending a large number of requests to a website server, causing it to slow down or crash.

2. DDoS Attack (Distributed Denial-of-Service):

- Multiple systems (often infected with malware and controlled as a botnet) are used to launch a coordinated attack on the target.
- Example: In 2016, a massive DDoS attack on DNS provider Dyn disrupted sites like Twitter, Netflix, and PayPal.

3.Impact:

- Service outages
- Financial losses
- Damage to reputation
- Potential entry points for further attacks

4. Prevention & Protection:

- Use firewalls and intrusion prevention systems (IPS)
- Deploy rate limiting and traffic filtering
- Use cloud-based DDoS protection services (e.g., Cloudflare, AWS Shield)
- Monitor network traffic for suspicious activity
- Have an incident response plan ready

Example: The 2016 Dyn DDoS attack used a botnet to bring down major websites like Twitter, Netflix, and PayPal.

5. Man-in-the-Middle (MitM) Attacks – Intercepting communications between two parties.

Definition:

A Man-in-the-Middle (MitM) attack occurs when a cybercriminal secretly intercepts and possibly alters communication between two parties who believe they are directly communicating with each other.

How It Works:

1. The attacker places themselves between the sender and the receiver.
2. They capture, eavesdrop, or even modify the data being exchanged.
3. The original parties are unaware their communication is being intercepted.

Common Techniques Used in MitM Attacks:
- **Wi-Fi Eavesdropping**: Setting up rogue Wi-Fi hotspots to intercept data.
- **Session Hijacking**: Stealing session tokens to impersonate a user.
- **HTTPS Spoofing**: Tricking users into visiting fake sites that mimic legitimate ones.
- **DNS Spoofing**: Redirecting users to malicious websites by altering DNS responses.

Example:
A user connects to a free, unsecured public Wi-Fi at a café. A hacker on the same network intercepts login credentials as the user logs into their bank account.

Prevention Measures:
- Use encrypted communication protocols (e.g., HTTPS, SSL/TLS).
- Avoid public Wi-Fi or use a VPN (Virtual Private Network).
- Enable Multi-Factor Authentication (MFA).
- Keep software and devices updated and patched.
- Be cautious of suspicious websites and unusual login prompts.

Cyber security Mechanisms to Prevent Hacking

1. Authentication and Access Control
- **Multi-Factor Authentication (MFA):** Adds an extra layer of security beyond passwords.
- **Role-Based Access Control (RBAC):** Restricts access based on user roles.

2. Data Protection and Encryption
- **End-to-End Encryption:** Secures data transmission against interception (e.g., SSL/TLS).
- **Data Masking:** Hides sensitive information from unauthorized users.

3. Network Security Measures
- **Firewalls:** Monitor and filter incoming and outgoing network traffic.
- **Intrusion Detection and Prevention Systems (IDPS):** Detect and block malicious activities.

4. Malware Protection
- **Antivirus and Anti-Malware Software:** Scans and removes malicious programs.
- **Sandboxing:** Isolates suspicious files in a controlled environment for analysis.

5. Security Awareness and Training
- **Employee Training:** Educates users on recognizing phishing and social engineering attacks.
- **Regular Security Audits:** Ensures vulnerabilities are identified and mitigated.

By implementing strong security mechanisms, organizations and individuals can effectively defend against hacking attempts and maintain a secure digital environment.

Preventive Measures Against Hacking
- **Use strong passwords** and enable multi-factor authentication (MFA).
- **Regularly update software** to fix security vulnerabilities.
- **Implement firewalls and intrusion detection systems** (IDS) to monitor and block cyber threats.
- **Educate users** about phishing scams and safe browsing practices.
- **Conduct regular security audits** and ethical hacking tests.

Hacking plays a dual role in cybersecurity—it can be a threat when used maliciously but also a powerful defense mechanism when used ethically. Organizations must stay proactive by implementing strong security measures and leveraging ethical hacking to protect against evolving cyber threats.

15.2. CRYPTOGRAPHY:

Cryptography is the practice of securing information by converting it into an unreadable format, ensuring confidentiality, integrity, and authenticity in cybersecurity. It plays a crucial role in protecting sensitive data from unauthorized access, cyber threats, and malicious attacks.

Figure.15.3 Cryptography.

15.2.1. Key Principles of Cryptography: They are 4 Types **.**

1. Confidentiality

- Ensures that only authorized users can access the information.
- Prevents data from being read by unauthorized individuals.
- Example: Encrypting an email so only the intended recipient can read it.

2. Integrity

- Guarantees that the data has notbeen altered or tampered with during transmission or storage.
- Example: A checksum or hash value confirms a file hasn't been modified.

3. Authentication

- Confirms the identity of users, devices, or systems.
- Ensures that the communication or data is from a legitimate source.
- Example: Logging in with a username, password, and OTP.

4. Non-repudiation

- Prevents individuals from denying their actions such as sending a message or performing a transaction.
- Provides proof of origin and integrity.
- Example: A digital signature ensures the sender can't deny sending a document.

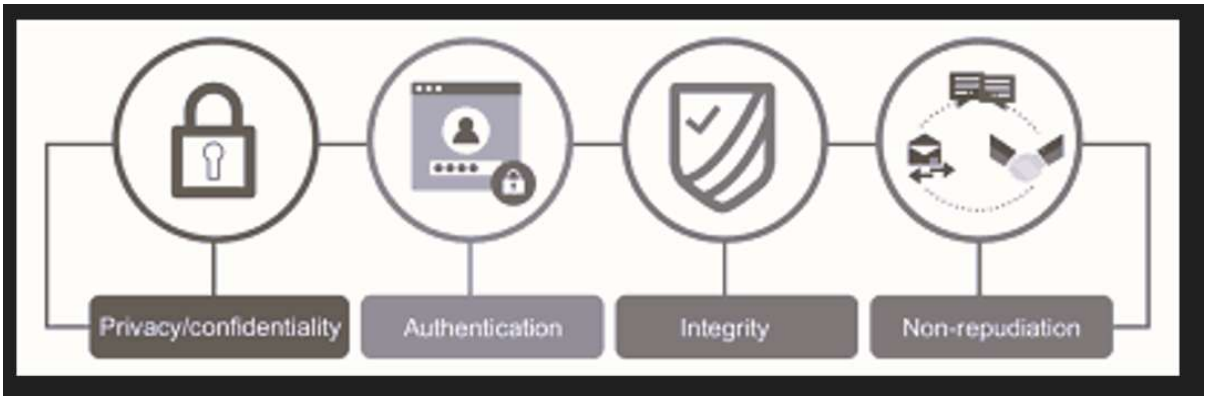


Figure.15.4Key Principles of Cryptography.

15.2.2. Types of Cryptography

1. Symmetric Cryptography (Secret Key Encryption)

- **Definition**: A cryptographic method that uses the same secret key for both encryption and decryption of data.
- **Key Feature**: Both sender and receiver must have access to the same key.
- **Speed**: Faster than asymmetric encryption; suitable for large volumes of data.
- **Security Concern**: Key distribution is challenging—if the key is intercepted, data security is compromised.

Examples:

- AES (Advanced Encryption Standard)
- DES (Data Encryption Standard)
- RC4, Blowfish

Common Applications:

- File and disk encryption
- Securing communication within a closed network
- Encrypting data stored in databases

2. Asymmetric Cryptography (Public Key Encryption)

- **Definition**: A cryptographic method that uses a pair of keys—a public key (shared) for encryption and a private key (secret) for decryption.
- **Key Feature**: Data encrypted with the public key can only be decrypted with the corresponding private key, and vice versa.
- **Security**: More secure for data transmission, as the private key is never shared.
- **Speed**: Slower than symmetric encryption; often used to encrypt small amounts of data (e.g., keys or hashes).

Examples:

- RSA (Rivest–Shamir–Adleman)
- ECC (Elliptic Curve Cryptography)
- DSA (Digital Signature Algorithm)

Common Applications:

- Secure email (PGP, S/MIME)
- Digital signatures
- SSL/TLS protocols for secure web browsing
- Blockchain and cryptocurrencies

3. Hash Functions in Cryptography

- **Definition**: A hash function is a mathematical algorithm that converts data into a fixed-length string of characters, called a hash value or digest.
- **One-way process**: It's computationally infeasible to reverse the hash back to the original data.
- **Deterministic**: The same input will always produce the same hash.

Key Properties:

- **Fixed Output Length**: No matter the size of input, output hash length is constant.
- **Pre-image Resistance**: Difficult to find the original input from its hash.
- **Collision Resistance**: Hard to find two different inputs that produce the same hash.
- **Avalanche Effect**: A small change in input causes a major change in output.

Common Hash Algorithms:

- SHA-256 (Secure Hash Algorithm 256-bit)
- SHA-1 (older, less secure)
- MD5 (outdated and insecure)

Applications:
- Data integrity checks
- Digital signatures
- Password hashing and storage
- Blockchain

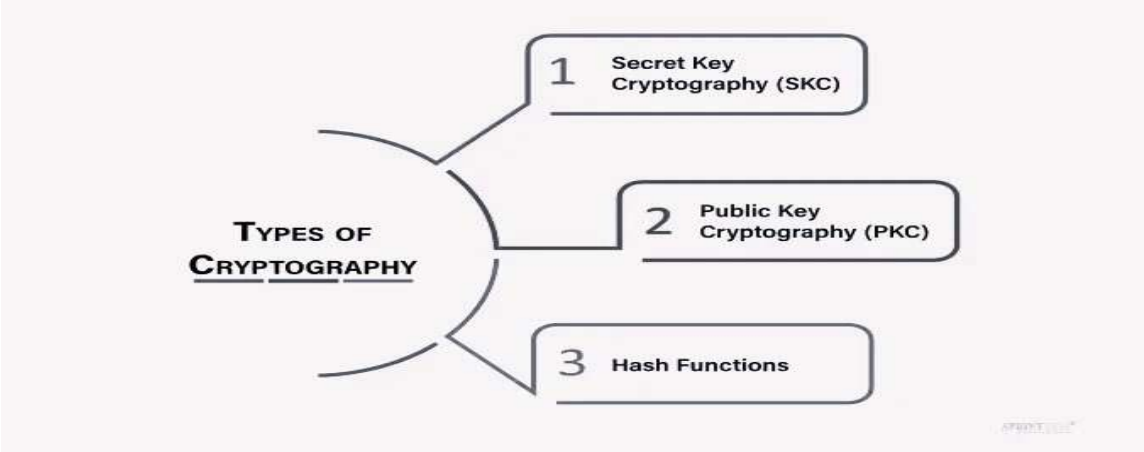


Figure.15.5 Types of Cryptography.

15.2.3. Applications of Cryptography in Cybersecurity
- **Data Encryption:** Protects sensitive data in storage and transit (e.g., SSL/TLS for secure web browsing).
- **Digital Signatures:** Ensures authenticity and integrity of documents and transactions.
- **Secure Authentication:** Enables multi-factor authentication (MFA) and password hashing.
- **Blockchain Security:** Provides tamper-proof records in decentralized systems.

Cryptography is fundamental to modern cybersecurity, safeguarding digital communications, financial transactions, and personal data from cyber threats.

15.3. DIGITAL SIGNATURES:

A digital signature is a cryptographic technique used to verify the authenticity, integrity, and origin of digital messages or documents. It ensures that data has not been altered during transmission and confirms the identity of the sender, making it a crucial security mechanism in cybersecurity.

15.3.1. How Digital Signatures Work
Digital signatures rely on asymmetric encryption, using a pair of cryptographic keys:
1. **Private Key:** Used by the sender to sign the document.
2. **Public Key:** Used by the receiver to verify the authenticity of the signature.

Steps in Digital Signature Process:

1. Key Generation
- A pair of keys is generated:
  - **Private Key**: Kept secret by the sender.
  - **Public Key**: Shared with others to verify the signature.

2. Creating a Digital Signature

- The sender writes the message or document.
- A hash function is applied to the data to generate a unique digest (hash value).
- This hash value is encrypted with the sender's private key**,** forming the digital signature**.**
- The original message and the digital signature are sent together to the recipient.

3. Signature Verification

- The recipient receives the message and the digital signature.
- The recipient:
  - Uses the sender's public key to decrypt the digital signature, obtaining the original hash.
  - Computes a new hash of the received message.
- If both hash values match:
  - The message is authentic and unaltered.
  - The signature is valid and sent by the claimed sender.

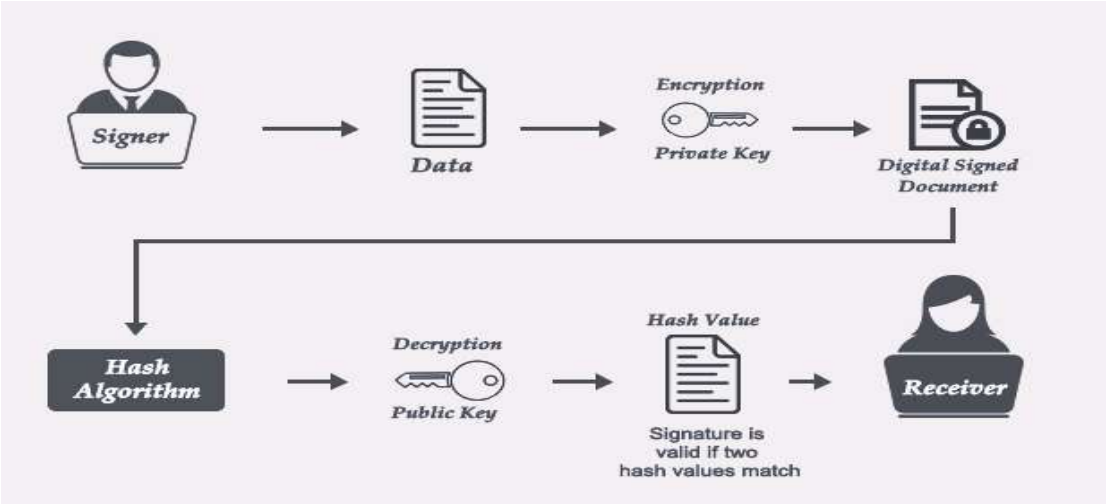


Figure.15.6 Digital signatures.

15.3.2. Process of Digital Signing and Verification

1. **Hashing** – The document or message is converted into a unique hash value using a hash function (e.g., SHA-256).
2. **Encryption** – The sender encrypts the hash using their private key, creating the digital signature.
3. **Transmission** – The document and digital signature are sent to the recipient.
4. **Verification** – The recipient decrypts the signature using the sender's public key and compares the hash to ensure authenticity and integrity.

15.3.3. Examples of Digital Signatures in Cybersecurity

1. **Secure Emails (S/MIME, PGP)**
   - Email services like Microsoft Outlook and Gmail use digital signatures to verify the sender's identity and prevent email spoofing.
   - Example: A company executive digitally signs an email before sending it to employees to ensure it is from an authentic source.
2. **Software Distribution and Code Signing**
   - Software vendors digitally sign applications to verify authenticity and prevent malware injection.
   - Example: Microsoft signs Windows updates with a digital signature to ensure they are from a trusted source.

3. **Online Banking and Financial Transactions**
   - Digital signatures secure online banking transactions, ensuring data integrity and preventing fraud.
   - Example: A user authorizes an online fund transfer using a digital signature, ensuring the transaction is legitimate.
4. **E-Governance and Legal Documents**
   - Governments use digital signatures for tax filings, digital IDs, and electronic contracts.
   - Example: The IRS in the U.S. accepts digitally signed tax returns to confirm authenticity.
5. **Blockchain and Cryptocurrency**
   - Blockchain transactions use digital signatures to validate ownership and prevent tampering.
   - Example: Bitcoin transactions require a digital signature to verify the sender's identity and authorize fund transfers.

Advantages of Digital Signatures

1. Authentication
- Verifies the sender's identity using cryptographic keys.

2. Integrity
- Ensures that the message/data has not been altered during transmission.

3. Non-Repudiation
- The sender cannot deny having sent the message, as only their private key could generate the signature.

4. Time-Saving and Efficient
- Speeds up document verification and approvals in digital workflows.

5. Cost-Effective
- Reduces paper, printing, and postage costs by enabling fully digital processes.

6. Legal Validity
- Recognized legally in many countries (e.g., under IT Act in India, eIDAS in the EU).

7. Environmental Friendly
- Reduces paper usage, supporting green and sustainable practices.

Disadvantages of Digital Signatures

1. Need for a Trusted Certificate Authority (CA)
- Requires a reliable third party (CA) to issue and manage digital certificates.

2. Private Key Compromise
- If a private key is stolen or leaked, the entire security system is at risk.

3. Complex Setup
- Involves cryptographic algorithms, certificates, and key management which can be complex for beginners.

4. Dependence on Technology
- Requires digital infrastructure like software, hardware, and internet access.

5. Legal and Regulatory Issues
- Not all countries accept digital signatures with the same legal weight.

6. Revocation and Expiry
- Digital certificates can expire or be revoked, making the signature invalid if not updated.

Digital signatures are essential in modern cybersecurity, ensuring trust and security in digital interactions across industries.

15.4 SUMMARY:

Hacking involves unauthorized access to computer systems, often exploiting security flaws. It can be ethical (white hat), malicious (black hat), or somewhere in between (grey hat). Ethical hacking helps organizations identify and fix vulnerabilities. Hacking has evolved from early experiments at MIT in the 1960s to sophisticated cyber threats today. Security mechanisms defend against hacking through authentication, encryption, and network protection. Common cyberattacks include phishing, malware, SQL injection, DDoS, and Man-in-the-Middle (MitM) attacks. Phishing tricks users into revealing sensitive information via fake messages or websites. Malware includes viruses, ransomware, spyware, and other malicious software.

SQL injection exploits web form vulnerabilities to manipulate databases. DoS/DDoS attacks flood systems with traffic, disrupting services. MitM attacks intercept and alter data between communicating parties. Preventive measures include firewalls, MFA, antivirus tools, and security training. Cryptography secures data using encryption techniques to ensure confidentiality and authenticity. Symmetric cryptography uses one shared key, while asymmetric uses public and private key pairs. Hash functions create fixed-length digests for data integrity and verification. Cryptography is vital in secure communications, online banking, and blockchain. Digital signatures validate the sender's identity and ensure data hasn't been tampered with. They use asymmetric cryptography—private keys to sign and public keys to verify. Applications include secure emails, legal e-documents, software updates, and financial transactions. Together, ethical hacking, cryptography, and digital signatures form the backbone of cyber security.

15.5. KEY TERMS:

Hacking, Cybersecurity, Ethical Hacking, Black Hat, Grey Hat, Phishing, Malware, SQL Injection, DDoS Attack, Encryption, Cryptography, Digital Signature, Public Key, Data Integrity, Authentication.

15.6 REVIEW QUESTIONS:

1. What is the difference between white hat, black hat, and grey hat hackers? Provide examples for each.
2. What are the key objectives of hacking security mechanisms in cybersecurity?
3. Explain how a phishing attack works and list at least three prevention tips.
4. Describe the types of malware and provide an example of a real-world malware attack.
5. How does a SQL Injection attack compromise a database, and what are the best practices to prevent it?
6. What are the main principles of cryptography and how do they contribute to cybersecurity?
7. Compare symmetric and asymmetric cryptography with suitable examples and use cases.
8. What is a hash function and why is it important in data integrity and security?
9. Describe the process of creating and verifying a digital signature.
10. List and explain three real-world applications of digital signatures in cybersecurity.

15.7 SUGGESTED READINGS:

1. "Cybersecurity: The Essential Body of Knowledge" – by Dan Shoemaker
2. "Principles of Information Security" – by Michael E. Whitman & Herbert J. Mattord
3. "Computer Security: Principles and Practice" – by William Stallings & Lawrie Brown
4. "Network Security Essentials: Applications and Standards" – by William Stallings
5. "The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" – by Dafydd Stuttard & Marcus Pinto
6. "Hacking: The Art of Exploitation" – by Jon Erickson
7. "Applied Cryptography: Protocols, Algorithms, and Source Code in C" – by Bruce Schneier

Dr. B. Basaveswara Rao

LESSON-16

FIREWALLS, AUTHENTICATION, AND CYBERSECURITY POLICIES

OBJECTIVES:

The objectives of the lesson are

1. To understand the critical components of network security—firewalls, authentication methods, and cybersecurity policies—and their role in safeguarding organizational systems and ensuring secure, compliant operations.
2. Explore the different types of firewalls and their role in defending against unauthorized access and cyber threats, ensuring secure network environments.
3. Learn to apply secure authentication mechanisms and create comprehensive cybersecurity policies that address access control, risk management, and regulatory compliance.

STRUCTURE:

16.1. Firewalls
- **16.1.1. What is a Firewall**
- **16.1.2. Design of Firewalls**
- **16.1.3. Types of Firewalls**

16.2. Firewalls Identification & Authentication with Biometric Techniques
- **16.2.1. Introduction to Firewalls**
- **16.2.2. Identification & Authentication in Firewalls**
- **16.2.3. Identification Techniques in Firewalls**
- **16.2.4. Authentication Techniques in Firewalls**
- **16.2.5. Role of Biometrics in Firewalls**
- **16.2.6. Challenges of Biometric Authentication in Firewalls**

16.3. Security Policies in Cybersecurity
- **16.3.1. Types of Security Policies**
- **16.3.2. Key Components of a Security Policy**
- **16.3.3. Importance of Security Policies**
- **16.3.4. Challenges in Implementing Security Policies**
- **16.3.5. Best Practices for Effective Security Policies**

16.4. Need for Legislation in Cybersecurity
- **16.4.1. Introduction**
- **16.4.2. Importance of Cybersecurity Legislation**
- **16.4.3. Key Cybersecurity Laws & Regulations**
- **16.4.4. Challenges in Implementing Cybersecurity Laws**

16.5. Cyber Laws in Cybersecurity
- **16.5.1. Introduction to Cyber Laws**
- **16.5.2. Importance of Cyber Laws**
- **16.5.3. Key Areas Covered Under Cyber Laws**
- **16.5.4. Major Cybersecurity Laws Around the World**
- **16.5.5. Challenges in Enforcing Cyber Laws**

16.6. Cybersecurity Issues
16.6.1. Introduction
16.6.2. Major Cybersecurity Issues
16.6.3. Prevention & Mitigation Strategies

16.7 Summary

16.8 Key Terms

16.9 Review Questions

16.10 Suggested Readings

16.1. FIREWALLS:

Firewalls were officially introduced in the early 1990s, but the underlying concept traces back to the reference monitor developed about two decades earlier.

16.1.1. What is a Firewall

A firewall is a device that filters all traffic between a secure, "inside" network and a less trustworthy, "outside" network. Typically, it operates on a dedicated device to ensure optimal performance, as it serves as the central point for traffic flow. To maintain efficiency, other non-firewall functions should not be performed on the same machine.

Since a firewall is executable code, it poses a potential security risk—an attacker could compromise the code and gain control over the firewall device. To minimize this risk, firewalls usually run on proprietary or carefully minimized operating systems, reducing the number of exploitable components.

The primary function of a firewall is to prevent unauthorized access and block harmful activities from entering a protected environment. To achieve this, firewalls enforce a security policy tailored to specific threats. For instance, a policy might block all external access while permitting internal traffic to pass outward. Alternatively, it could allow access only from specific sources, users, or for designated activities. A key challenge in firewall protection is determining the appropriate security policy that best aligns with the organization's needs.

16.1.2. Design of Firewalls:

A reference monitor must meet the following criteria:

- **Always invoked** – It must be activated for every access attempt.
- **Tamperproof** – It should be resistant to unauthorized modifications.
- **Small and simple** – It must be minimal and straightforward to allow rigorous analysis.

A firewall serves as a specialized type of reference monitor. When strategically placed within a network, it ensures that all controlled network access passes through it, fulfilling the "always invoked" requirement. Firewalls are typically well-isolated, often implemented on a dedicated machine with direct connections only to internal and external networks, enhancing their resistance to tampering and meeting the "tamperproof" standard. Additionally, firewall designers emphasize maintaining a simple and focused functionality to ensure effectiveness and security.

16.1.3. Types of Firewalls:

Firewalls can be categorized based on their functionality, deployment, and architecture. Below are the main types of firewalls:

1. Based on Filtering Method

a) Packet Filtering Firewalls

- Operate at the network layer (Layer 3) of the OSI model.
- Examine data packets and allow or block them based on predefined rules (IP address, ports, protocols).
- Example: Access Control Lists (ACLs) in routers.

b) Stateful Inspection Firewalls

- Operate at the network and transport layers (Layers 3 & 4).
- Keep track of active connections and make filtering decisions based on the connection state.
- More secure than packet filtering firewalls.

c) Proxy Firewalls (Application Layer Firewalls)

- Operate at the application layer (Layer 7).
- Act as intermediaries between users and the internet, inspecting traffic at the application level.
- Offer deep packet inspection and content filtering.
- Example: Squid Proxy.

d) Next-Generation Firewalls (NGFWs)

- Combine traditional firewall capabilities with advanced security features like intrusion prevention, malware detection, and deep packet inspection.
- Can identify and control applications instead of just filtering ports and protocols.
- Example: Palo Alto Networks, Cisco Firepower.

2. Based on Deployment Location

a) Network Firewalls

- Deployed at the perimeter of a network to monitor incoming and outgoing traffic.
- Can be hardware-based or software-based.

b) Host-Based Firewalls

- Installed on individual devices (laptops, servers) to monitor and control traffic to and from that specific device.
- Example: Windows Defender Firewall, iptables (Linux).

c) Cloud Firewalls (Firewall-as-a-Service, FWaaS)

- Hosted in the cloud to protect cloud-based applications and services.
- Provides scalable security for distributed networks.
- Example: AWS Firewall, Cloudflare Firewall.

3. Based on Architecture

a) Hardware Firewalls

- Standalone physical devices that monitor and filter network traffic.
- Typically used in enterprises and data centers.

b) Software Firewalls

- Installed on an operating system and configured to filter network traffic.
- Suitable for personal computers and servers.

c) Virtual Firewalls

- Software-based firewalls deployed in virtual environments like cloud infrastructure or virtual machines.

- Used in cloud security and virtualized data centers.

Each type of firewall serves a specific purpose, and organizations often use a combination of these firewalls for enhanced security.

16.2. FIREWALLS IDENTIFICATION & AUTHENTICATION WITH BIOMETRIC TECHNIQUES:

16.2.1. Introduction to Firewalls

A firewall is a network security device that monitors and controls incoming and outgoing traffic based on predefined security rules. It acts as a barrier between a trusted internal network and untrusted external networks.

16.2.2. Identification & Authentication in Firewalls

Identification and authentication mechanisms in firewalls ensure that only authorized users and devices can access the network. These mechanisms help in preventing unauthorized access, cyberattacks, and data breaches**.**

16.2.3. Identification Techniques in Firewalls

Identification is the process of recognizing a user or device before granting access. Common identification techniques include:

1. **Username/ID-Based Identification** – Users provide a unique identifier, such as a username or employee ID.
2. **IP Address-Based Identification** – The firewall verifies the device's IP address before allowing traffic.
3. **MAC Address Filtering** – The firewall allows or blocks devices based on their unique MAC address.
4. **Digital Certificates** – Public Key Infrastructure (PKI) is used for identification via cryptographic certificates.

16.2.4. Authentication Techniques in Firewalls

Authentication ensures that the identified user or device is legitimate. Common authentication techniques include:

a) Traditional Authentication Methods

1. **Password-Based Authentication** – Users enter a password to verify their identity.
2. **Two-Factor Authentication (2FA)** – Requires a second verification step, such as a One-Time Password (OTP).
3. **Multi-Factor Authentication (MFA)** – Uses multiple verification factors like a password, OTP, and biometrics.

b) Biometric Authentication Techniques

Biometric authentication is a highly secure method that verifies users based on their unique biological traits. Common biometric techniques used in firewalls include:

1. **Fingerprint Recognition** – Scans and verifies the user's fingerprint before granting access.
2. **Facial Recognition** – Uses AI to match the user's face with stored biometric data.
3. **Iris Scanning** – Scans the unique patterns of the iris for authentication.
4. **Voice Recognition** – Identifies users based on their voice patterns.
5. **Behavioral Biometrics** – Analyzes user behaviors like keystroke dynamics and mouse movements.

16.2.5. Role of Biometrics in Firewalls

- **Enhanced Security** – Eliminates risks associated with stolen passwords or credential-based attacks.
- **User Convenience** – No need to remember passwords; authentication is quick and seamless.
- **Prevention of Spoofing** – Difficult to forge or duplicate biometric data.
- **Adaptive Security** – Some firewalls integrate AI-driven biometric authentication for real-time threat detection.

16.2.6. Challenges of Biometric Authentication in Firewalls

- **Privacy Concerns** – Biometric data is sensitive and needs to be stored securely.
- **False Positives/Negatives** – Errors in biometric recognition can either grant or deny access incorrectly.
- **Cost & Infrastructure** – Implementing biometric authentication in firewalls requires additional hardware and software.
- **Data Storage & Processing** – Large-scale biometric authentication requires secure and efficient data storage solutions.

Firewalls play a crucial role in network security, and incorporating biometric authentication enhances their ability to prevent unauthorized access. While biometric techniques offer high security and convenience, organizations must address privacy concerns, infrastructure costs, and potential recognition errors for successful implementation.

16.3. SECURITY POLICIES IN CYBERSECURITY:

A security policy is a formal set of rules and guidelines that define how an organization protects its data, network, and IT infrastructure from cyber threats. Security policies help ensure confidentiality, integrity, and availability (CIA Triad)of information.

16.3.1. Types of Security Policies

A. Organizational Security Policies

High-level policies that define an organization's security objectives and responsibilities.

- **Information Security Policy** – Establishes the framework for protecting company data.
- **Acceptable Use Policy (AUP)** – Defines acceptable usage of IT resources like the internet, email, and devices.
- **Access Control Policy** – Regulates who can access what data, based on roles and responsibilities.
- **Data Classification Policy** – Categorizes data as public, internal, confidential, or highly sensitive.

B. System-Specific Security Policies

Guidelines for securing specific IT systems such as firewalls, databases, and servers.

- **Network Security Policy** – Defines firewall rules, VPN usage, and network monitoring practices.
- **Email Security Policy** – Regulates email usage, spam filtering, and phishing protection.
- **Cloud Security Policy** – Specifies security measures for cloud storage, access, and data sharing.

C. Issue-Specific Security Policies

Address specific cybersecurity concerns such as BYOD (Bring Your Own Device) or incident response.

- **Password Policy** – Enforces password complexity, expiration, and multi-factor authentication (MFA).
- **Remote Access Policy** – Defines VPN usage, remote desktop protocols, and secure authentication for offsite employees.
- **Incident Response Policy** – Outlines steps to follow in case of a security breach or cyber attack.
- **Backup and Recovery Policy** – Ensures regular data backups and disaster recovery planning.

16.3.2. Key Components of a Security Policy

1. **Purpose** – The objective of the policy and why it is needed.
2. **Scope** – Defines who and what is covered (employees, contractors, third parties).
3. **Roles & Responsibilities** – Specifies duties of security teams, IT staff, and users.
4. **Security Controls** – Specifies technical and procedural safeguards.
5. **Compliance Requirements** – References legal and industry regulations (e.g., GDPR, ISO 27001).
6. **Enforcement & Penalties** – Outlines consequences for policy violations.

16.3.3. Importance of Security Policies

1. **Prevents Cyber Threats** – Protects against malware, phishing, ransomware, and data breaches.
2. **Ensures Compliance** – Helps meet legal and regulatory requirements (HIPAA, PCI-DSS).
3. **Protects Sensitive Data** – Prevents unauthorized access to critical business information.
4. **Improves Incident Response** – Provides clear guidelines for handling security incidents.
5. **Enhances Employee Awareness** – Educates staff about best security practices.

16.3.4. Challenges in Implementing Security Policies

1. **Lack of Awareness** – Employees may not follow security policies if not properly trained.
2. **Complexity** – Overly complicated policies can be difficult to enforce.
3. **Evolving Threats** – Cybersecurity policies need regular updates to address new threats.
4. **Non-Compliance** – Employees or third-party vendors may fail to comply with security rules.

16.3.5. Best Practices for Effective Security Policies

1. **Keep Policies Clear & Concise** – Avoid technical jargon and ensure easy understanding.
2. **Regularly Update Policies** – Review and modify security policies as threats evolve.
3. **Conduct Security Training** – Educate employees on best security practices.
4. **Monitor& Enforce Policies** – Implement monitoring tools and enforce compliance.
5. **Align with Industry Standards** – Follow best practices from frameworks like NIST, ISO 27001, CIS Controls**.**

Note: Security policies are the backbone of cybersecurity strategy. They establish rules, responsibilities, and security controls to safeguard an organization's digital assets**.** Implementing clear, well-structured policies enhances data protection, regulatory compliance, and risk management**.**

16.4. NEED FOR LEGISLATION IN CYBERSECURITY:

16.4.1. Introduction

Cybersecurity legislation is essential to protect individuals, organizations, and governments from cyber threats, data breaches, and cybercrimes. With the increasing reliance on digital systems, strong laws are needed to regulate online activities, ensure data privacy, and penalize cybercriminals.

16.4.2. Importance of Cybersecurity Legislation

1. **Protection Against Cyber Threats**
   - Prevents cyberattacks such as hacking, phishing, malware, and ransomware.
   - Safeguards critical infrastructure like banking, healthcare, and government networks.
2. **Data Privacy & Protection**
   - Ensures personal and sensitive information is not misused or accessed without authorization.
   - Enforces data protection standards for businesses handling customer data.
3. **Legal Framework for Cybercrimes**
   - Defines cybercrimes such as identity theft, online fraud, and hacking.
   - Establishes penalties for cybercriminals to deter illegal activities.
4. **National Security & Defense**
   - Protects government data from cyber espionage and attacks by foreign entities.
   - Strengthens cybersecurity policies for military and intelligence agencies.
5. **Regulation of Businesses & Organizations**
   - Mandates security measures for companies handling financial transactions.
   - Requires businesses to report data breaches and implement cybersecurity best practices.
6. **International Cooperation**
   - Helps countries collaborate on cross-border cybercrime investigations.
   - Ensures a global response to threats like cyberterrorism and ransomware attacks.

16.4.3. Key Cybersecurity Laws & Regulations

- **General Data Protection Regulation (GDPR)** – Protects user data and privacy in the EU.
- **Cybersecurity Information Sharing Act (CISA, USA)** – Promotes information sharing between the government and private sector.
- **Personal Data Protection Bill (India)** – Regulates data protection and privacy in India.
- **Computer Fraud and Abuse Act (CFAA, USA)** – Criminalizes unauthorized access to computer systems.
- **ISO 27001** – International standard for information security management systems.

16.4.4. Challenges in Implementing Cybersecurity Laws

Rapidly Evolving Threats – Cybercriminals constantly develop new attack methods.
Lack of Awareness – Many individuals and businesses are unaware of cybersecurity laws.
Jurisdiction Issues – Cybercrimes often involve multiple countries, making prosecution difficult.
Balancing Security & Privacy – Striking a balance between security enforcement and user privacy.

Note: Cyber security legislation is crucial in today's digital world to protect individuals, businesses, and governments from cyber threats. Strong laws ensure data privacy, crime prevention, national security, and regulatory compliance**.**Continuous updates and global cooperation are needed to tackle evolving cyber risks effectively.

16.5. CYBER LAWS IN CYBERSECURITY:

16.5.1. Introduction to Cyber Laws

Cyber laws refer to the legal regulations and policies that govern online activities, digital transactions, and cybersecurity. These laws are designed to protect individuals, organizations, and governments from cybercrimes such as hacking, identity theft, data breaches, and online fraud.

16.5.2. Importance of Cyber Laws

Protects Personal & Financial Data – Prevents unauthorized access to sensitive information.
Defines & Punishes Cybercrimes – Establishes legal consequences for cyber offenses.
Regulates Online Activities – Ensures responsible and lawful use of the internet.
Enhances National Security – Prevents cyberterrorism and protects critical infrastructure.
Encourages Safe E-Commerce – Protects consumers and businesses in online transactions.

16.5.3. Key Areas Covered Under Cyber Laws

1. **Data Protection & Privacy Laws**
   - Regulate how personal and financial data is collected, stored, and shared.
   - Example: General Data Protection Regulation (GDPR - EU)**,** Personal Data Protection Bill (India)**.**
2. **Cybercrime Laws**
   - Define cyber offenses like hacking, phishing, cyberstalking, and identity theft.
   - Example: Computer Fraud and Abuse Act (CFAA - USA)**,** Information Technology (IT) Act (India).
3. **Intellectual Property (IP) Rights & Copyright Laws**
   - Protect digital content, software, and intellectual property from piracy and unauthorized use.
   - Example: Digital Millennium Copyright Act (DMCA - USA)**.**
4. **E-Commerce & Online Transaction Laws**
   - Ensure security in online banking, digital payments, and electronic contracts.
   - Example: Electronic Signatures in Global and National Commerce Act (E-SIGN - USA)**.**
5. **Cybersecurity & National Security Laws**
   - Protect governments, military, and businesses from cyber espionage and cyber warfare.
   - Example: Cybersecurity Information Sharing Act (CISA - USA)**.**

16.5.4. Major Cybersecurity Laws Around the World.

Country/Region	Cyber Law	Purpose
USA	Computer Fraud and Abuse Act (CFAA)	Criminalizes unauthorized computer access
EU	General Data Protection Regulation (GDPR)	Protects user data and privacy
India	Information Technology (IT) Act, 2000	Regulates cybercrimes and e-commerce
UK	Computer Misuse Act, 1990	Addresses hacking and cyber offenses
China	Cybersecurity Law of China	Regulates internet security and data protection
Australia	Cybercrime Act, 2001	Defines cyber offenses and penalties

16.5.5. Challenges in Enforcing Cyber Laws

Cross-Border Jurisdiction Issues – Cybercrimes often involve multiple countries, making prosecution difficult.
Rapidly Evolving Cyber Threats – Laws need constant updates to address new cyber risks.
Privacy vs. Security Debate – Finding a balance between user privacy and cybersecurity enforcement.
Lack of Awareness – Many users and organizations are unaware of cyber laws and regulations.
Cyber laws play a vital role in **protecting individuals, businesses, and governments** from cyber threats. As cyber risks evolve, continuous updates to legislation and international cooperation are essential for a safer digital world.

16.6. CYBER SECURITY ISSUES:

16.6.1. Introduction

Cybersecurity issues refer to threats, vulnerabilities, and risks that affect the security of digital systems, networks, and data. These issues can lead to data breaches, financial losses, identity theft, and disruption of critical infrastructure.

16.6.2. Major Cybersecurity Issues

A. Cyber Threats & Attacks

1. **Malware (Viruses, Worms, Trojans, Ransomware)**
   - Malicious software designed to harm or exploit systems.
   - **Example:** Ransomware encrypts data and demands payment for decryption.
2. **Phishing Attacks**
   - Fraudulent emails or messages that trick users into revealing sensitive information.
   - **Example:** Fake banking emails asking for login details.
3. **Denial-of-Service (DoS) & Distributed Denial-of-Service (DDoS) Attacks**
   - Overloads a website or network, making it unavailable to users.
   - **Example:** Botnets attacking online services.
4. **Man-in-the-Middle (MitM) Attacks**
   - Intercepting communication between two parties to steal or manipulate data.

- **Example:** Unsecured public Wi-Fi snooping.

5. **SQL Injection Attacks**
   - Exploiting database vulnerabilities to access or delete sensitive information.
   - **Example:** Attackers injecting malicious SQL code in login forms.
6. **Zero-Day Exploits**
   - Attacks that exploit unknown vulnerabilities before a fix is available.
   - **Example:** Cybercriminals exploiting software flaws before patches are released.
7. **Insider Threats**
   - Employees or contractors misusing access privileges to steal or compromise data.
   - **Example:** Disgruntled workers leaking confidential information.

B. Data Security Issues

1. **Data Breaches**
   - Unauthorized access to sensitive data, leading to financial and reputational damage.
   - **Example:** Facebook and LinkedIn data leaks.
2. **Identity Theft & Fraud**
   - Stealing personal information for financial gain or impersonation.
   - **Example:** Credit card fraud using stolen personal details.
3. **Weak Passwords & Credential Leaks**
   - Using simple passwords or reusing them across multiple platforms increases risk.
   - **Solution:** Enforce strong passwords and Multi-Factor Authentication (MFA).
4. **Cloud Security Risks**
   - Misconfigured cloud storage leading to data exposure.
   - **Example:** AWS S3 bucket misconfigurations exposing sensitive files.

C. Emerging Cybersecurity Issues

1. **IoT Security Threats**
   - Poorly secured Internet of Things (IoT) devices can be hacked.
   - **Example:** Smart home devices used in botnet attacks.
2. **Artificial Intelligence (AI)-Powered Attacks**
   - Hackers using AI for sophisticated phishing, deepfake scams, and automated hacking.
3. **Deepfake Technology**
   - AI-generated videos or audio used for misinformation, fraud, or impersonation.
   - **Example:** Fake political speeches or identity fraud.
4. **Quantum Computing Risks**
   - Future quantum computers could break traditional encryption.
5. **Cyberwarfare & Nation-State Attacks**
   - Governments engaging in cyber espionage and attacks on other countries.
   - **Example:** Stuxnet malware targeting Iran's nuclear program.

16.6.3. Prevention & Mitigation Strategies

- **Use Strong Passwords & Multi-Factor Authentication (MFA)** – Reduces unauthorized access.

- **Keep Software & Systems Updated** – Patch vulnerabilities to prevent exploits. **Implement Firewalls & Intrusion Detection Systems (IDS)** – Enhances network security.
- **Regular Data Backups** – Protects against ransomware and accidental data loss.
- **User Awareness & Training** – Educate employees on phishing, social engineering, and security best practices.
- **Use Encryption for Data Protection** – Prevents unauthorized data access. **Secure Cloud & IoT Devices** – Configure security settings and use strong authentication.

Cybersecurity issues are continuously evolving, requiring constant vigilance, strong security measures, and global cooperation. Organizations and individuals must stay proactive in protecting digital assets, personal data, and critical infrastructure from cyber threats.

16.7 SUMMARY:

Firewalls are crucial devices that filter traffic between secure and untrusted networks, aiming to prevent unauthorized access and harmful activities. Introduced in the 1990s, firewalls work by enforcing security policies, such as blocking external access or controlling network traffic.

They can be categorized by filtering method (packet filtering, stateful inspection, proxy, NGFWs), deployment location (network, host-based, cloud), and architecture (hardware, software, virtual). Identification and authentication techniques, including biometric methods like fingerprint and facial recognition, enhance firewall security. Security policies, which establish rules for protecting data and IT systems, are essential in cybersecurity. They address threats, ensure compliance, and guide incident responses. Cybersecurity legislation is critical for protecting against cyber threats, data breaches, and cybercrimes, with laws like GDPR and CISA offering frameworks for regulation. The continuous evolution of cybersecurity challenges demands regular updates and international cooperation.

16.8 KEY TERMS:

Firewall, Security policy, Authentication, Biometric authentication, Packet filtering, Stateful inspection, Proxy firewall, Next-generation firewall, Cloud firewall, Cybersecurity legislation, Data protection, Cybercrime, Denial-of-service, Malware, Identity theft.

16.9 REVIEW QUESTIONS:

1. What is the primary function of a firewall in network security?
2. Explain the concept of "reference monitor" and how it relates to firewalls.
3. Describe the different types of firewalls based on their filtering methods. Provide an example for each type.
4. How do biometric authentication techniques enhance the security of firewalls? List at least three biometric methods used for authentication.
5. What are the key components of a security policy in cybersecurity?
6. Why is cybersecurity legislation important, and how does it protect organizations from cyber threats?
7. Explain the role of cybersecurity laws in protecting personal and financial data. Provide examples of major cybersecurity laws in different countries.

16.10 SUGGESTED READINGS:

1. Stallings, W. (2017).Network Security Essentials: Applications and Standards *(*6th ed.). Pearson.
2. Williamson, K., & Tarrant, J. (2015).Firewalls and Internet Security: Repelling the Wily Hacker(2nd ed.). Addison-Wesley.
3. Cole, E., & Ring, L. (2019).Hacking Exposed*:* Network Security Secrets & Solutions
4. Linn, J., & Sumner, M. (2016).Cybersecurity: The Beginner's Guide(1st ed.). McGraw-Hill Education.
5. Schneier, B. (2015).Secrets and Lies: Digital Security in a Networked World(2nd ed.). Wiley.

Dr. B. Basaveswara Rao