Yashwantrao
Chavan
Maharashtra
Open University

BBA 2076

Computer Application

Author: Rajashri Sonawane

YASHWANTRAO CHAVAN MAHARASHTRA OPEN UNIVERSITY, NASHIK.

VICE-CHANCELLOR: PROF.SANJEEV SONAWANE

PRO-VICE-CHANCELLOR: DR. JOGENDRASINGH BISEN

DIRECTOR, SCHOOL OF COMMERCE AND MANAGEMENT- DR. SURENDRA PATOLE

Author

Rajashri Sonawane
Brahmdevdada Mane Institute of Technology,
Solapur

Editor

Dr. Prashant Suresh Salve
B. A. College, Pathardi,
Ahmednagar

Instructional Technology Editing and Programme Co-ordinator

Dr. Surendra Patole
Director,
School of commerce and Management
YCMOU, Nashik-422222.

Production

Shri. Vilas Badhan
Head, Printing and Production center, YCMOU, Nashik

Yashwantrao Chavan Maharashtra Open University, Nashik
(*First edition of developed under DEB development grant*)

- First Publication: Publication Code:
- Type Setting: Cover Design:
- Printed by:
- Publisher:

ISBN NO: BBA 2076

3.2.3 Interpreters
3.2.4 Editors
3.3 Introduction to Scripting Languages

UNIT 4: Programming in C++

4.1 Introduction to C++
4.2 Basic Syntax and Structure
4.3 Variables and Data Types
4.4 Control Structures (if-else, loops)
4.5 Functions and Arrays
4.6 Object-Oriented Programming Concepts

BLOCK III: SOFTWARE ENGINEERING AND USER INTERFACE DESIGN

UNIT 5: Measures, Metrics, and Indicators in Software Engineering

5.1 Importance of Measures and Metrics
5.2 Common Software Metrics
5.3 Indicators in Software Engineering
5.4 Software Quality Assurance

UNIT 6: User Interface Design in Software Engineering

6.1 Principles of User Interface Design
6.2 User-Centered Design
6.3 Prototyping and Usability Testing
6.4 Human-Computer Interaction

BLOCK IV: MICROSOFT OFFICE APPLICATIONS

UNIT 7: Introduction to MS Word

7.1 Creating Simple Documents
7.2 Editing Text
7.3 Working with Tables and Graphics
7.4 Formatting Documents
7.5 Using Tools (Spell-Check, Hyphenation, Mail-Merge)
7.6 Printing Documents, Envelopes, and Labels
7.7 Document Collaboration Features

UNIT 8: Advanced Features of MS Word

8.1 Styles and Templates
8.2 Macros and Automation
8.3 Collaboration Features (Track Changes, Comments)
8.4 Advanced Table Features
8.5 Advanced Document Formatting

UNIT 9: Introduction to MS Excel

9.1 Understanding Workbooks and Worksheets
9.2 Opening and Saving Workbooks
9.3 Basic Formulas and Functions
9.4 Formatting Cells and Data
9.5 Data Sorting and Filtering

UNIT 10: Advanced Features of MS Excel

10.1 Creating and Printing Workbooks/Sheets
10.2 Using Advanced Formulas and Functions
10.3 Creating Graphs and Charts
10.4 Data Analysis Tools (PivotTables, Data Validation)
10.5 Conditional Formatting and Advanced Charting

UNIT 11: Introduction to MS PowerPoint

11.1 Creating Presentations
11.2 Inserting and Formatting Text and Images
11.3 Using Slide Layouts and Themes
11.4 Adding Animations and Transitions
11.5 Basics of Slide Design

UNIT 12: Advanced Features of MS PowerPoint

12.1 Creating Master Slides
12.2 Incorporating Multimedia (Audio, Video)
12.3 Advanced Animation Techniques
12.4 Delivering and Printing Presentations
12.5 Interactive Presentations

BLOCK V: ADVANCED TOPICS IN COMPUTING

UNIT 13: Database Management Systems (DBMS)

13.1 Introduction to DBMS
13.2 Types of Databases
13.3 Database Models
13.4 SQL Basics and Applications
13.5 Database Design Principles

UNIT 14: Cybersecurity and Data Protection

14.1 Introduction to Cybersecurity
14.2 Common Cyber Threats and Attacks
14.3 Data Protection Techniques
14.4 Best Practices in Cybersecurity
14.5 Legal and Ethical Issues in Cybersecurity

UNIT 15: Cloud Computing

15.1 Introduction to Cloud Computing
15.2 Types of Cloud Services (IaaS, PaaS, SaaS)
15.3 Benefits and Challenges of Cloud Computing
15.4 Applications of Cloud Computing
15.5 Cloud Security and Compliance

UNIT 16: Emerging Technologies in Computing

16.1 Artificial Intelligence (AI) and Machine Learning (ML)
16.2 Internet of Things (IoT)
16.3 Blockchain Technology
16.4 Future Trends in Computing
16.5 Quantum Computing

Unit 1: Introduction to Computers and Data Representation

Learning Outcomes:

- Students will be able to define the binary number system.
- Students will be able to describe the process of converting binary to decimal and vice versa.
- Students will be able to perform binary arithmetic operations.
- Students will be able to apply Boolean algebra in binary arithmetic.
- Students will be able to analyze the differences and applications of various computer codes.

Structure:

1.1 Binary Number System

1.1.1 Conversion of Binary to Decimal and Vice-Versa

1.1.2 Binary Arithmetic

1.2 Introduction to Boolean Algebra

- Knowledge Check 1
- Outcome-Based Activity 1

1.3 Computer Codes

1.3.1 BCD (Binary Coded Decimal)

1.3.2 ASCII (American Standard Code for Information Interchange)

1.3.3 EBCDIC (Extended Binary Coded Decimal Interchange Code)

1.3.4 Unicode

1.3.5 Parity Codes

1.4 Evolutionary Software Process Models

1.4.1 The Incremental Model

1.4.2 The Spiral Model

- Knowledge Check 2
- Outcome-Based Activity 2

1.5 Overview of Computer Architecture

1.6 Summary

1.7 Keywords

1.8 Self-Assessment Questions

1.9 References / Reference Reading

1.1 Binary Number System

Number systems are used in digital electronics to represent and manipulate numerical data. The most commonly used number systems are binary, decimal, and hexadecimal. Binary is the most fundamental number system in digital electronics, representing digital signals in computers and electronic devices. Decimal is the number system most commonly used in everyday life, while hexadecimal is often used in computer programming and digital electronics.

1.1.1 Conversion of Binary to Decimal and Vice-Versa

For putting down the value of a number, most number systems follow the same pattern: A predetermined number of values can be represented with a single numerical character, followed by a new column that counts how many times the greatest value in the counting system has been attained. The number of numerical values that the system employs is referred to as the system's base. The decimal system, for example, includes ten number characters and so a base of 10:

0 1 2 3 4 5 6 7 8 9

When writing numerals bigger than 9, a second column is added to the left, with 10 times the value of the column to its right.

Because the base values of the number systems typically utilised in digital electronics differ from those of the decimal system, they appear less recognisable yet function roughly the same.

Decimal, (base 10)

Decimal, also known as base 10, is a numerical system that uses ten digits to represent numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9.

Each digit in a decimal number represents a power of 10. For example, in the number 235, the digit 5 represents 5 units, the digit 3 represents 3 groups of 10 units, and the digit 2 represents 2 groups of 100 units.

Binary, (base 2)

Binary, also known as base 2, is a numerical system that uses only two digits, 0 and 1, to represent numbers.

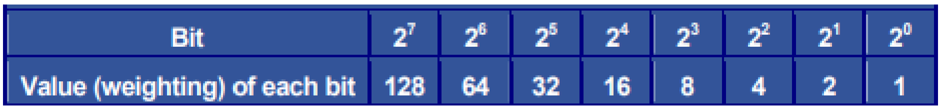
For instance:

10_{10} denotes the decimal value ten. (1 ten + 0 units)

10_2 denotes the binary value two. (1 two + 0 units)

Converting Binary to Decimal

To convert from binary to decimal, write down the binary number with the correct 'weighting,' i.e. the value of the columns, starting with one for the right-hand (least important column or LEAST SIGNIFICANT BIT) column. As you proceed left, give each column double the value of the preceding column.



Bit	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Value (weighting) of each bit	128	64	32	16	8	4	2	1

Example:

To convert the binary number 01000102 to decimal, follow these steps:

1. Write down the binary number and assign a weight to each bit based on its position.
2. Sum the values of the columns that have a bit of 1, ignoring the columns with a bit of 0.

Applying the appropriate weighting to0100010 gives $(0 \times 2^6) + (1 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) = (34)_{10}$

Therefore: $0100010_2 = 34_{10}$

How to Convert Decimal Numbers to Binary Numbers

There are several ways they can be converted, such as the division method.

Here is how to convert a decimal number to binary using the division method: This is a procedure on how to convert decimal number to binary using division method:

1. Divide the Decimal Number by 2: Now it is time to divide the given decimal number with the divisor ‘2’. Let us record the quotient and the remainder.
2. Determine Remainder for Even Numbers: If the even decimal number is divided by one, it will give you another whole figure, and the remainder will be zero.
3. Determine Remainder for Odd Numbers: When the decimal number is odd, the division is not rounded off to zeroes, and the quotient contains one recurring decimal figure of the number one.
4. Record the Remainders: Each time you do a divide, write down the remainders beginning from the first division you did first. When aligning it to get the binary value, arrange it in order, starting from the LSB at the top and MSB at the bottom.

1.1.2 Binary Arithmetic: Binary arithmetic is one of the basic subjects that pupils are expected to learn about digital electricity and calculation. It involves performing arithmetic operations using the binary number system, which consists only of two digits. Regarding the randomised parameter, the attribute has two possible states, which include 0 and 1. This system is applied in computation in digital computers in the same way that the decimal system is used in computers.

Binary Number System: In the binary system of numbering, each digit forms a power of 2 and starts with the digit at the right as equal to 2 zero. For example, the binary number 1101 can be converted to decimal as follows: For instance, converting the binary number 1101 to decimal, we get the following:

$$ 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 0 + 1 = 13 $$

Basic Binary Arithmetic Operations

1. Binary Addition

Binary addition is similar to decimal addition, though there are fewer rules because there are only two digits. The rules are:

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0 = 1
- 1 + 1 = 10 (which is 0 with a carry of 1)

Example:

```
  1011
+ 1101
------
 11000
```

2. Binary Subtraction

Binary subtraction also employs borrowing in the same way it is done in the decimal system. The rules are:

- 0 - 0 = 0
- 1 - 0 = 1
- 1 - 1 = 0
- 0 - 1 = 1 (with a borrow of 1)

Example:

```
  1011
- 0101
------
  0110
```

3. Binary Multiplication

Binary multiplication is straightforward, following the same principles as decimal multiplication, with simpler rules: Binary multiplication is straightforward, following the same principles as decimal multiplication, with simpler rules:

- 0 × 0 = 0
- 0 × 1 = 0
- 1 × 0 = 0
- 1 × 1 = 1

Example:

```
  101
×  11
-----
  101
+1010
------
 1111
```

4. Binary Division

It is similar to that for long division in decimals that involve comparison and repeated subtraction. It follows these steps:

- Place the divisor on the leftmost digits of the dividend.
- Especially since sometimes you can compare and, in this case, subtract some numbers.
- In this case, move the digit over to the right side and perform the above steps again.

Example:

1101 ÷ 10 = 110 R 1

1.2 Introduction to Boolean Algebra

Boolean algebra is a branch of higher mathematics that helps to study and control the changes in logical expressions. It deals with variables that can take on two possible values: represented by 1 when the statement is true and 0 when the statement is false. Incorporated by the mathematician who was called George Boole, Boolean algebra is one of the most significant branches in the field of digital electronics, computer science, and logic design. Here is an introduction to Boolean algebra:

1. **Variables:** In Boolean algebra, three types of variables are used to represent logical propositions or conditions. These variables can only have two possible values: in outcome, ‘true’ can be equal to 1 and ‘false’ can be equal to 0. For instance, a variable (A) can stand for “It is raining. ”
2. **Logical Operators:** Boolean algebra divides several logical operators that work on variables into four main groups. The basic operators include:
   - AND ((and)): Stands for the joining of two statements. The given result can be true only if both propositions are true.
   - OR ((or)): Used to depict a logical connection of a proposition and the negation of a proposition. This is true inasmuch as at least the truth of one of the propositions under consideration is sufficient to warrant the resultant truth.
   - NOT ((not)): Denotes the not of a proposition. It negates it and changes its truth value to the opposite of the given proposition.
3. **Truth Tables:** Truth tables depict how logical operators operate, taking into consideration all feasible possibilities of the input variables. The rows of the truth table represent all potential input variable conditions, and the column represents the output of the operation with the current logical operator.
4. **Boolean Expressions:** Boolean expressions are algebraic terms that are formed from variables and also logical operators. They present logical relationships and conditions in a compact, organised manner regularly. For instance, (A and B) means the joining of proposition (A) and proposition (B).
5. **Properties of Boolean Algebra:** Boolean algebra follows a number of algebraic laws, and these are:
   1. Commutative: The first two formulated equations are (A and B = B and A) and (A or B = B or A).
      - Associative: The associative order is ((A and B) and C = A and (B and C)) and ((A or B) or C = A or (B or C)).
   2. Distributive: (A and (B or C) = (A and B) or (A and C)) and (A or (B and C) = (A or B) and (A or C))

- **Knowledge Check 1**

Fill in the blanks

1. In binary arithmetic, each digit is called a ___. (bit)

2. According to the addition rule between binary numbers, if one adds one to one, the result in binary is ___, with a carry of __. (10, 1)
3. Supposing that there is a subtraction of a larger digit from a smaller digit, it involves borrowing, for instance, ___-1. (0)
4. In binary multiplication, each digit of the multiplier is multiplied by each digit of the ___. (multiplicand)

- **Outcome-Based Activity 1**

  To convert the binary number 010010_2 to decimal and vice-versa.

1.3 Computer Codes

Computer codes are the techniques of encoding characters, symbols with digital form, and control codes. Every code gives number sequences to the various characters, which makes it possible for computers to display text and other information formats. Here's a brief overview of some commonly used computer codes:

1.3.1 BCD (Binary Coded Decimal)

BCD, or Binary-Coded Decimal, is a system of encoding decimal numbers in binary form. In BCD, each decimal digit is represented using four bits in binary form. For instance, the decimal number 123 will be represented by the following: BCD 0001 0010 0011.

1.3.2 ASCII (American Standard Code for Information Interchange)

ASCII, also known as the American standard code for information interchange, is the most common character encoding standard. It utilises 7 or 8 bits to represent text characters. Developed for the English language, ASCII assigns numbers to each character, including letters, digits, punctuation signs, and control signs. For example, the ASCII value of the capital letter A is 65.

1.3.3 EBCDIC (Extended Binary Coded Decimal Interchange Code)

EBCDIC is a coded character set that IBM invented for use in mainframe computers. EBCDIC, more than ASCII, has 8 bits and contains some characters compatible with IBM mainframes.

1.3.4 Unicode

Unicode is a character encoding specification that strives to be used to support all the text characters of all the world’s current and potential writing scripts. It has a code that allocates each character, alphabets, number symbols, emojis, and special signs. There are many forms of Unicode, and a few of them are UTF-8, UTF-16, and UTF-32.

1.3.5 Parity Codes

Parity is a technique used to detect errors in transmitted data. In parity checking, an additional bit (the parity bit) is added to the data to ensure that the total number of bits set to 1 in the data, including the parity bit, is always odd (odd parity) or even (even parity). An error is detected if the number of bits set to 1 does not match the expected parity.

1.4 Evolutionary Software Process Models

The evolutionary software process models are the effective models of the software development process that comprise the iteration and increments as a development approach. These models presuppose that requirements and solutions are not static and try to incorporate change at any stage of development. Here are some notable evolutionary software process models:

1. Prototyping Model:

- The Prototyping Model focuses on first constructing a basic form of the software (prototype) in order to receive responses for fine-tuning the demands.
- Concerning the end user's feedback, the user iterates the prototype in order to get the final product that needs to be developed.

2. Incremental Model:

- The Incremental Model breaks the software development process into incremental steps and is mostly used when the development process is large and complex.
- Each increment provides a part of the functionality and is to be delivered and reviewed by the stakeholders.
- Gradually, the previous incarnations also acquired new features and innovations that were added from time to time.

3. Iterative Model:

- The Iterative Model divides the construction of software into cycles or iterations.
- Each cycle consists of requirements analysis, design, implementation, testing, and deployment cycle.
- Ideas from one cycle feed into the other, making the process build on itself; in other words, feedback is used to make adjustments.

4. Agile Model:

- Methodologies such as Scrum, Kanban, and Extreme Programming are known to have their major focuses on flexibility, collaboration, and customer involvement.
- Scrum teams work in a particular period called sprints and repeatedly provide a new functioning product in small periods.
- Agile focuses on values like working on change rather than the plan and helps place teams at a point where they can be able to work according to the changing needs and demands of clients and markets.

5. Spiral Model:

- The second type of SDLC, the Spiral Model, integrates iterative models with risks and prototypes.
- Testing and again planning, risk appraisal, modelling, building and assessment are included as repeatable cycles.
- The latter is important to note since each Cycle to be mentioned as spirals underlines the previous one and encompasses feedback and the proposed risks

These evolutionary software process models provide changeable strategies for the development of software to facilitate teams in managing risks and changes and efficiently producing high-quality software. The option of a particular model depends on the state of affairs of the project, the team, the customer, and other conditions.

1.4.1 The Incremental Model

The Incremental Model is a model of software development that divides the development process into small principles or increments. Each increment provides a part of the functionality, enabling the delivery of the required solution to stakeholders in portions and receiving feedback. The Incremental Model follows a series of steps. It includes:

1. **Requirements Gathering:**

- The general necessities of the software system are identified and aggregated.
- The requirements are then arranged according to their value and the achievability of implementation.

2. **Initial Planning:**

- A coarse-grained plan to define the project's extent and the increments is established.

3. **Increment Development:**

- It starts with the first increment, which lies in the development of the first set of features of the system.

4. **Increment Delivery:**

- Stakeholders are given an increment that has been developed and tested and asked to give their views on it.

5. **Feedback Incorporation:**

- It also involves incorporating the changes that the users in the software have identified to improve it.

6. **Iterative Development:**

- The development process is cyclical, and each subsequent increment is added to the existing structure.

7. **Incremental Integration:**

- When new increments are being developed, they are built to work with the previously delivered increments in order to support compatible and coordinated systems.
- This is done to ensure that the increments are fully integrated since integration testing is the process used to test the integrated software components.

8. **Final Delivery:**

- The incremental model reaches its final stage in the delivery of the final increment, which contains all the proposed features and improvements.
- The developed software system is deployed for use in production or delivered to the user, consumer or client.

1.4.2 The Spiral Model

This can also be best described as the software development process model that incorporates the risk management technique as well as the prototyping in stages. Developed by Barry Boehm in 1986, this model is more suitable for large and complex software development projects with high risks and a high level of unpredictability. The development process is in the form of a spiral, which means that each cycle depicts a phase of the lifecycle. Here are the key characteristics and phases of the Spiral Model:

1. **Planning:**

- Learning: Define the objectives of the project, as well as the constraints and risks that are likely to be encountered.
- Identify the prospects of the project and draw a rough working plan.

2. Risk Analysis:

- It is necessary to define and evaluate the possible risks of the project implementation.
- Risks are to be assessed and ranked depending on their consequences and their probability.
- Formulate the techniques on how to handle risk factors and contingency plans.

3. Engineering:

- The steps include the following: acquire, collate, and record the requirements of the software.
- Design and build the system structure.
- Develop prototypes to assess the so-defined and designed system requirements.

4. Evaluation:

- Discuss with the team members and evaluate the accomplishments made and the software's readiness and quality.
- Ask stakeholders and the end-users to point out what they think needs to be changed.

- **Knowledge Check 2**

  Fill in the Blanks

  1. The Prototyping Model involves the development of ___ of the software in order to obtain feedback. (prototypes)
  2. Scrum and Kanban are some of the most used Agile methodologies that focus on ___ rather than on schedules. (deliverables)
  3. The Iterative Model basically discusses the development process in terms of ___ cycles or iterations. (repeated)
  4. The Spiral Model integrates features of iterative development with ___ management and prototyping. (risk)

- **Outcome-Based Activity 2**

  Create a PowerPoint presentation on Evolutionary Software Process Models.

1.5 Overview of Computer Architecture

Computer architecture is a concept that describes computer systems, the designs implemented in the hardware constituents of computers, and how they are integrated to perform a particular function. It combines the management of the central processing unit (CPU), memory, input/output (I/O) devices, and communication between them.

Computer architecture is a field that deals with designing computing systems to be faster, more efficient, and scalable. This involves developing processors that can perform instructions at high speeds, managing data storage and access mechanisms within memory, and establishing other control mechanisms between the several components of the computer's hardware.

Some of the concepts of computer architecture are instruction set architecture (ISA), which is the set of instructions that a CPU can execute, and micro-architecture, which is the implementation of ISA using hardware components like registers, arithmetic and logic units, and control unit.

Computer architects must compromise aspects like performance, power usage, costs, and compatibility while designing computer systems. Technological enhancements and the emergence of new computing models and structures always lead to research in computer organisation to build better, quicker, and more powerful systems.

1.6 Summary

- Number systems are mainly employed in digital electronics to denote or work with number values.
- There are three known number systems, namely, binary, decimal, and hexadecimal. Binary is the simplest form of the number system used in digital electronics since it is used to carry out digital signals on computers and other digital equipment.
- Boolean algebra is the algebraic technique used in the formal analysis of logic expressions. It deals with variables that can take on two possible values: Real numbers that are assigned to the Boolean logic include true, represented by the numeric digit one (1) and false, represented by the numeric digit zero (0).
- Computer codes are special forms of encoding that are used to represent characters, symbols, and control codes on computers. Each code sets specific binary patterns to various characters so that computers can analyse and send text and other information.
- Evolutionary software process models are more of a life cycle approach to developing software and are characterised by flexibility and iteration.

- The Incremental Model is a software development technique that divides a project into small and more manageable phases or iterations.
- The Spiral Model is a SDP process model that combines iteration, risk management, and prototyping following a specific pattern. This model was formulated by Barry Boehm in 1986 and is most useful when the software development project is immense and unpredictable.

1.7 Keywords

- **Spiral Model:** This is the development of software that incorporates risk management, iteration and prototyping.
- **Boolean algebra:** It is a branch of mathematics that deals with the logical calculation of propositions or statements
- **Computer codes:** Computer codes are sets of symbols used to represent characters, symbols, and control codes in the digital format.

1.8 Self-Assessment Questions

1. Explain the concept of place value in the binary number system.
2. Provide a detailed description of how to perform multiplication and division operations in binary.
3. Explain with examples what Boolean algebra is and its role in the design of digital logic circuits.
4. Explain what computer codes are and what they do concerning characters and symbolisation.
5. ASCII, EBCDIC, and Unicode are some of the most used encoding schemes; briefly explain their differences.

1.9 References / Reference Reading

- Mano, Morris M., and Charles R. Kime. *Logic and Computer Design Fundamentals*. 4th ed., Pearson, 2008.
- Rajaraman, V., and T. Radhakrishnan. *An Introduction to Digital Computer Design*. 5th ed., PHI Learning, 2018.
- Floyd, Thomas L. *Digital Fundamentals*. 11th ed., Pearson, 2014.

- Balagurusamy, E. *Fundamentals of Computers*. 6th ed., McGraw Hill Education, 2014.
- Bartee, Thomas C. *Digital Computer Fundamentals*. 6th ed., McGraw Hill, 1991.

Unit 2: Problem Analysis and Algorithm Design

Learning Outcomes:

- Students will be able to identify steps in problem analysis.
- Students will be able to explain the importance of algorithms in problem-solving.
- Students will be able to create flowcharts using standard symbols.
- Students will be able to differentiate between various software testing techniques.
- Students will be able to design pseudocode for given problems.

Structure:

2.1 Problem Analysis

One of the major components of problem-solving in computer science is understanding and defining the problem one wants to solve before finding a solution. This stage is quite important in software development and system design processes. Here's a detailed look at the key components and steps involved in problem analysis:

1. Problem Definition

- Clear Statement of the Problem: First, the problem should be defined in its simplest terms to help develop understanding. Make sure all the stakeholders have a clear understanding of the problem.
- Objectives and Goals: Specify the objectives of the solution, or in other words, what do you want to accomplish with it? As part of the planning, it follows the specifics of the objectives as well as key success factors.

2. Requirements Gathering

- Functional Requirements: Determine the specific tasks for the system. This could involve data processing or the interactions of the users with the systems and the subsequent responses.
- Non-functional Requirements: Some of these include performance, security, usability, and reliability. These are critical because they help prove that the system is functional in the actual environment.
- Stakeholder Input: Involve users, clients and developers in order to elicit all necessary information from them.

3. System Context and Constraints

- System Boundaries: State what the system will cover and what will not be included in the system. This reduces cases of scope inflation and guarantees developmental concentration.
- Constraints: Determine any constraints in advance, such as technology constraints, budget constraints, time constraints, legal constraints, and constraints of the current system.

4. Use Case and User Story Templates

- Use Cases: Define scenarios/freeware descriptions to explain how the system will be utilised. In each use case, the description should be created from the user's point of view and should include the actions related to a particular function.
- User Stories: Develop use case scenarios to record specific requirements from the perspective of the user. These are formed in the structure of 'Given [user], I want to [act] in order to [accomplish an objective].'

5. Data Requirements

- Data Inputs: Determine the input data that the system will need. Determine where data is going to be collected and how it will be obtained or captured.
- Data Outputs: Identify the outcomes that the system should generate: What kind of reports, notifications, and other outputs are expected?
- Data Storage and Management: Consider how data will be saved, accessed, sorted out, and even deleted. This involves aspects of DB design such as, data organisation and protection issues as well as data integrity.

6. Analysis Models

- Diagrams and Models: Employ flowcharts, ERDs, class diagrams, and any other model that will be useful in describing the problem and the intended solutions.
- Process Flows: Audit the processes and activities that are performed to determine system performance and inefficiencies.

7. Risk Analysis

- Identify Risks: Enumerate possible threats that might affect the project including technical issues, shortages of resources, and organisational opposition.
- Risk Mitigation: Formulate actions to control the identified risks, which include having action plans in case of specified risk and preventive measures.

8. Documentation and Review

- Documentation: Develop a detailed record of all the problem analysis and findings, requirements, models, and feasibility studies.
- Review and Validation: Carry out interviews with the stakeholders to confirm the findings of the problem analysis. Ensure that all the specified requirements are credible, covering all the necessary aspects, and have been mutually understood by the stakeholders.

2.2 Algorithm Design

Algorithm design is one of the key computer science concepts dedicated to inventing algorithms for particular tasks. It involves elements of both critical thinking, brainstorming and calculations to arrive at the solution. Here’s a detailed guide on the key components and steps involved in algorithm design:

1. Understanding the Problem

- Problem Definition: Describe the situation before its improvement. When identifying the problem at hand, an important step is to describe the situation before the solution is applied. It is important to clarify the input, the expected results and the limitations.

- Examples and Edge Cases: It is crucial to think about regular instances and situations that are as close to the extremes as possible to understand the problem thoroughly and in all its aspects.

2. Analysing the Problem

- Input Size and Constraints: Determine the size of input data and limitations on time and space. This assists in selecting an appropriate strategy.
- Properties of the Problem: Check whether the problem to be solved has characteristics, for example, whether it is an optimal substructure or has overlapping subproblems this informs you whether to implement dynamic programming or greediness.

3. Choosing an Approach

- Brute Force: Be precise and focused to identify straightforward strategies that consider all of them. Good for rather small inputs but less suitable for larger data sets, for various reasons.
- Divide and Conquer: Divide the problem into simpler problems, solve the subordinate problems and then integrate the solutions. Some are merge sort and quick sort.
- Greedy Algorithms: Make the best decision at the local level and always expect to get the best overall solution. Appropriate for problems such as knapsack problems and some graph problems.
- Dynamic Programming: Use when the problem can be divided into several similar problems that are more manageable than the original problem. This is because while solving a problem, solutions to subproblems are also generated, and when solving another problem that has similar subproblems, it is always better to retrieve the solutions from the database and use them than to retrieve them. Some of the real-life applications are the Fibonacci sequence and the shortest path.
- Backtracking: Try every possible move and return to the previous state when it turns out to be nonvalid. VERY useful in CSPs like N-Queens problem and Sudoku.
- Heuristic and Approximation: Employ when it is impossible to obtain an exact solution to the given problem. They give adequate solutions within a reasonable time.

4. Designing the Algorithm

- Pseudocode: The algorithm should be written in pseudocode in order to explain the strategy clearly and briefly. This is useful in the visualisation of the steps and the structure.
- Data Structures: Select proper data structures that enable fast operations on data and easy access to data. For instance, arrays, linked lists, hash tables, trees, and graphs.
- Algorithm Steps: To implement the algorithm it is essential to divide it into certain easily understandable steps. Make sure that from one step you go to the next one and the links between the steps are reasonable.

5. Optimisation

- Improving Efficiency: Try to avoid having a high time and space complexity. This might mean improving loops, employing better data structures, or employing higher-level algorithms.
- Avoiding Redundancy: Reduce redundant calculations and use memoisation, or improve the logic to avoid repeating calculations.

6. Implementation

- Coding: Convert the pseudocode into the actual code of a programming language. Concentrate on such things as variable names, comments, and coding style.
- Testing: Thoroughly check its efficiency in several cases, including the worst and the best ones, and use various inputs.

7. Verification and Validation

- Proof of Correctness: Also, make use of mathematical proofs or formal approaches to make sure the algorithm responds to the problem in the right way.
- Empirical Testing: Check the correctness of the algorithm with the help of some sample data for which the correct result is known and test the algorithm on the real dataset.

8. Documentation

- Clear Description: Explain what the algorithm is, why it is used and what the process of algorithm steps looks like.
- Complexity Analysis: Record the time and space complexity analysis.
- Usage Instructions: State how the algorithm is to be used, the input format of data and what result is expected.

9. Review and Refinement

- Code Review: Allow other students to look at the work done on the algorithm and at its implementation to point out mistakes and such.

- Refinement: Reapply the algorithm with feedback and the results of the testing as well as performance metrics acquired.

2.2.1 Advantages and Disadvantages of Algorithms

Advantages of Algorithms

1. Efficiency:

- Time Efficiency: Complex algorithms can take a lot of time before they solve a certain problem but that time can be reduced by coming up with a well-designed algorithm. More efficient algorithms enable the program to process large datasets and solve more complicated problems in less time.

2. Reusability:

- Modularity: Algorithms are usually developed to be of general forms that may be used in different programs due to modularity. This modularity makes code maintainability and reusability a lot easier.

3. Scalability:

- Handling Large Inputs: Ideal algorithms can go up a couple of notches in complexity while managing the same amount of input data.

4. Predictability and Reliability:

- Consistent Results: Algorithms are very reliable in their operations because they offer the same solution for the same problem.

5. Automation:

- Reduced Human Effort: Computational methods can efficiently use the sequences, requiring less human input and fewer errors.

Disadvantages of Algorithms

1. Complexity:

- Design Complexity: Finding better algorithms is another big challenge because this process is very time-consuming. It involves knowledge of the problem and the principles of designing the algorithm to solve the problem at hand.

2. Resource Intensive:

- High Computational Resources: Certain algorithms, especially if they are to solve intricate challenges, are likely to be resource-intensive and not feasible for low-power gadgets.

3. Specialisation:

- Domain-Specific Algorithms: There are algorithms that can only be used to solve specific problems and cannot be used for other problems. This makes them somewhat useful in one area but not very much in other areas.

4. Assumptions and Constraints:

- Rigid Assumptions: It can be noted that most of the time algorithms are designed under certain assumptions such as the input is sorted, a certain data structure is used etc. If these assumptions do not hold, the algorithm may fail or fail.

5. Development Time

- Long Development Cycles: Algorithms development and tuning may take a long time and may slow down project completion.

- **Knowledge Check 1**

  Fill in the Blanks

  1. ___________ in computer science involves understanding and defining the problem that needs to be solved before designing a solution. (Problem analysis)
  2. _____________ involves a combination of problem-solving skills, creativity, and mathematical rigor. (Algorithm design)
  3. ___________ is the one of the Advantage of Algorithms. (Efficiency)
  4. Disadvantages of Algorithms are ________ & ___________. (complexity & time consumption)

- **Outcome-Based Activity 1**

  Discuss on the advantages & disadvantages of algorithms in class room

2.3 Flowcharts

Flowcharts are diagrams that can be used to illustrate the processes or algorithms that are inherent in a system. It is used in computer science, business and engineering to reduce comprehension and interpretational problems and aid in the communication of information and ideas. Here’s a detailed overview of flowcharts:

- **Components of a Flowchart**

1. Start/End (Terminator):

- Symbolised with the oval sign.
- Refs to the starting and or ending phases of a process.

2. Process (Action or Operation):

- It is illustrated by the rectangle.

  - Defines a stage where an action is done.

3. Decision:

- The shapes used to represent them include;
- Signals that a decision is made and depending on yes/no or true/false the program goes to a new branch.

4. Input/Output:

- Symbolised by a parallelogram.
- Shows where data is inputted into or outputted from the process.

5. Flow Lines (Arrows):

- Represented by arrows.
- Indicate the direction of flow from one step to another.

6. Connector:

- Represented by a small circle or a labeled circle.
- Used to connect flowchart segments too large to fit on one page or simplify complex diagrams.

7. Sub-process (Predefined Process):

- Represented by a rectangle with double lines on each side.
- Denotes a set of actions that are defined elsewhere, allowing for modularity and reuse.

- **Creating a Flowchart**

1. Define the Process:

- Clearly define the process or algorithm you want to represent.
- Identify the start and end points, major steps, decisions, inputs, and outputs.

2. List the Steps:

- List all the steps involved in the process in chronological order.
- Include any decisions that need to be made and the possible outcomes of each decision.

3. Choose Flowchart Symbols:

- Select the appropriate symbols for each step, decision, input, and output.
- Ensure you use standard symbols to maintain clarity and consistency.

4. Draw the Flowchart:

- Begin with the start symbol and connect each subsequent step using arrows.

- Clearly label each step and decision.
- Use decision symbols to split the flow based on outcomes, ensuring each branch is properly connected.

5. Review and Refine:

- Check the flowchart to see if it is mapped correctly in the process.
- Ensure no steps are missing, the correct symbols are used, and the arrows are positioned correctly.
- Explain where measures are needed to make it easier for people to read.

2.3.1 Symbols Used in Flowcharts

Many standardised symbols are used to represent the action, decision, and process in flowcharts. Here's a detailed look at the common symbols used in flowcharts:

Common Flowchart Symbols

1. Terminator (Start/End)

- Shape: Oval or Pill Shape
- Description: Points to the beginning as well as the end of the process.
- Usage: They are the starting and the finishing point of the flowchart.

2. Process (Action or Operation)

- Shape: Rectangle
- Description: Is a process, task, or operation.
- Usage: Applied to describe a stage whereby certain computations or alterations are carried out on some data.

3. Decision

- Shape: Diamond
- Description: Is a kind of decision point that enables the flow to split based on a yes/no or true/false question.
- Usage: Ends with a question and arrows that point to the various directions depending on the answer.

4. Input/Output

- Shape: Parallelogram
- Description: Shows the flow of data either fed into the process or produced by the process.
- Usage: Employed to indicate where data is input or where the results are obtained.

5. Flow Lines (Arrows)

- Shape: Arrows
- Description: Label each arrow to show from which step one is coming from and going to another step.
- Usage: Join symbols to illustrate the steps in the manner of sequence.

6. Connector

- Shape: Small Circle or Labeled Circle
- Description: Links one part of a flowchart to the other, commonly used where the flowchart is lengthy and may be extended to several pages.
- Usage: It assists to prevent drawing lines and makes the flowchart clean.

2.4 Coding and Testing

Coding and testing are essential activities that are carried out during the software development life cycle SDLC. They make sure that an application created meets a given need and functions as it is supposed to and as optimally as possible.

Coding (Implementation)

# 1. Coding Standards

- Consistency: Compliance with these standards means that the code is easily understandable, and any developer can easily work on the code.

# 2. Choosing the Right Programming Language

- Suitability: Choose a language that will be perfect for the project needs concerning Web development, system programming, mobile apps, etc.

# 3. Code Development

- Modularisation: Divide the given application into much smaller sub-modules or functions that can be tackled easily. This makes the code easier to write, debug, and update as required in the future also the project is easy to scale.
- Version Control: Suggested to use version control systems such as Git that help with managing changes to the codebase and the history of changes and collaborating with other team members.

# 4. Code Documentation

- Inline Comments: Be sure to add commentaries on the code; these will enable other persons to understand the logic you applied in the code.
- External Documentation: For an application, write an informative paper containing information about the general layout, a detailed description of the modules, the API reference, and how to use it.

# 5. Code Reviews

- Peer Review: Peer code review: Other members of the development team should review the code to find problems and share good practices.

Testing

# 1. Types of Testing

Unit Testing:

- Focus: Checks the units or subparts of the software.
- Tools: Java: JUnit; Python: pytest; .NET: NUnit.
- Automation: Generally done automatically to guarantee that the individual unit operates effectively on its own.

Integration Testing:

- Focus: This is a test of the interaction of different modules or components that are present in the software.
- Purpose: Checks that partially completed sections of the application function as anticipated with full sections.
- Tools: JUnit (Java), pytest (Python), TestNG.

System Testing:

- Focus: Involves testing the total system end-to-end.
- Scope: Confirms that the whole application complies with the stated standards.
- Tools: There are a few tools that are mostly used: Selenium, JMeter, and LoadRunner.

Acceptance Testing:

- Focus: Verify the software meets the user specifications.
- Types: User Acceptance Testing (UAT) and Business Acceptance Testing (BAT) are other similar tests that are performed at the end of the software development life cycle.
- Tools: Cucumber, FitNesse.

2.4.1 Software Testing Techniques

2.4.1.1 White-Box Testing

White-box testing, also known as clear-box or glass-box testing, involves testing an application's internal features. The tester has complete information about the code and can, as such, conduct comprehensive control and data flow testing and check for logic.

# Key Concepts in White-Box Testing

Techniques

1. Code Coverage:

- Types:

- Statement Coverage: This guarantees that each statement in a given code runs through the computer at least once.
- Branch Coverage: Check for coverage of each branch of the code to guarantee that every decision has been executed at least once.
- Path Coverage: Makes sure that every possibility or the dynamic path of the code is tested.
- Condition Coverage: Makes sure that each condition in the decision statements is tested to a true state as well as to a false state.

2. Unit Testing:

- Focus: Verification of individual entities of the software or sub-modules of the software.
- Tools: Java, JUnit; C#, NUnit; Python, pytest.

3. Control Flow Testing:

- Objective: Verifies that the program is logically constructed and will run from start to completion.
- Techniques:
- Control Flow Graphs (CFG) are Diagrams that depict all possible routes that might be followed in a program when it is running.

4. Data Flow Testing:

- Objective: Analyzes the movement of data through the program.
- Techniques:
- Def-Use Chains: Defines the variables and the progressive uses of the identified definitions.

2.4.1.2 Basis Path Testing

Basis Path Testing is a white box testing technique whose main goal is to test each code path. It assists in defining all the executable paths and guarantees that the paths are run at least once.

# Steps in Basis Path Testing

1. Create a Control Flow Graph (CFG): Create a Control Flow Graph (CFG):

- This means that we should represent the program’s control structure using nodes (representing processing tasks) and edges (representing control flow).

2. Calculate Cyclomatic Complexity:

- Cyclomatic complexity (CC) is a metric that can be used to determine the quantitative measure of the logical complexity of the program.
- Formula: \(V(G) = E - N + 2 \)
- \(E \): This is the total number of arcs in the CFG of the language.
- \(N \): Total nodes in the CFG.

3. Determine Independent Paths:

- The number of independent paths through the program should be determined. Independent paths are the paths that add at least one edge, which is not present on any other path.

4. Design Test Cases:

- To create test cases, it is necessary to ensure that at least one passage of each independent path is tested.

2.4.1.3 Cyclomatic Complexity

Cyclomatic Complexity is a metric used to measure the complexity of a program. It indicates the number of linearly independent paths through a program's source code. It helps determine the number of test cases needed for thorough testing.

# Calculation of Cyclomatic Complexity:

1. Using Control Flow Graph (CFG):

- Formula: \(V(G) = E - N + 2 \)
- \(E \): Number of edges (transitions between nodes).
- \(N \): Number of nodes (decision points).

2. Alternative Formula (for simpler structures):

- \(V(G) = D + 1 \)
- \(D \): Number of decision points in the program (e.g., if statements, loops).

# Interpretation:

- Complexity Levels:

- 1-10: Simple program, low risk.
- 11-20: Moderate complexity, moderate risk.
- 21-50: Complex, high risk.
- >50: Very complex, very high risk.

2.4.2 Control Structure Testing

Control structure testing is a white-box testing technique that focuses on testing a program's control flow. It involves designing test cases to ensure that all possible control flow paths within the program are exercised at least once. This technique aims to evaluate decision points (e.g., conditions, loops) and ensure that the program behaves as expected under different conditions.

Key Concepts in Control Structure Testing:

1. Control Flow Graph (CFG):

- A graphical representation of the control flow within a program.
- Nodes represent program statements, and edges represent possible transitions between statements.
- Helps visualize all possible paths through the program.

2. Nodes:

- Represent specific program statements or decision points (e.g., if statements, switch cases, loops).
- Each node in the CFG corresponds to a specific point in the program where control flow decisions are made.

3. Edges:

- Represent the possible transitions between nodes.
- Arrows connecting nodes indicate the flow of control from one statement to another.

4. Decision Coverage:

- A measure of the percentage of decision outcomes (true/false) that the test cases have exercised.
- Ensures that all decision points in the program have been evaluated under both true and false conditions.

- **Knowledge Check 2**

  Fill in the Blanks.

  1. __________ are visual diagrams that represent the steps and decision points within a process or algorithm. (Flowcharts)
  2. The oval in the flowchart is represented by ________. (start/end)
  3. White-box testing, also known as ________. (glass-box testing)

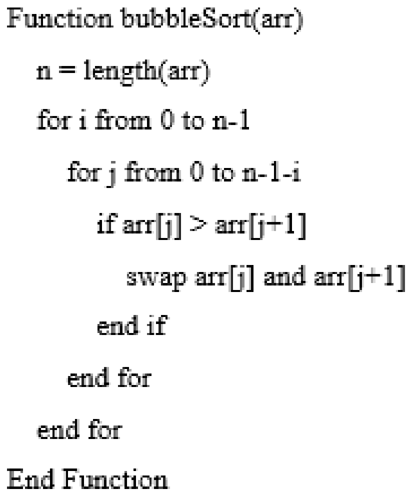
4. Control structure testing is a ________ testing technique. (white-box)

- **Outcome-Based Activity 2**

  Make a PowerPoint presentation on the symbols used in the flowchart.

2.5 Pseudocode and Its Applications

Pseudocode is a high-level description of an algorithm or a program, written in plain language and not bound by specific syntax rules of any particular programming language. It is the stage that follows the problem-solving stage but is done before the coding stage. Here's an example of pseudocode for a simple sorting algorithm:



```
Function bubbleSort(arr)
  n = length(arr)
  for i from 0 to n-1
    for j from 0 to n-1-i
      if arr[j] > arr[j+1]
        swap arr[j] and arr[j+1]
      end if
    end for
  end for
End Function
```

Applications of pseudocode include:

1. **Algorithm Design:** Pseudocode is beneficial in that it helps one describe the algorithm to be used in the actual programming before one starts the process of coding. It assists in comprehending the process that is carried out in solving a particular task.
2. **Teaching and Learning:** To sum up, pseudocode should be used in teaching as it enables the learners to concentrate on problem-solving without becoming overwhelmed by language specifics.
3. **Documentation:** Pseudocode can be used to comment on how algorithms and programs are designed and what they do. It gives a more general perspective that is less technical than code.
4. **Collaboration:** When using or creating a team, pseudocode can act as a universal language for communication and planning of algorithms and the program's logical flow, especially in the case of different programming language preferences.

2.6 Summary

- Problem analysis in computer science refers to the process of ascertaining and describing the problem that requires a solution before designing one. This is a very important phase in software development as well as in system design.
- Detailed requirements, constraints analysis, and feasibility assessment help teams create systems that fit users' needs and organisational objectives, minimise risks, and work within the constraints.
- Algorithm design can be described as a highly systematic and iterative activity that presupposes profound problem analysis and meticulous planning and validation.
- Process maps are helpful in the analysis of processes and are a vital part of designing and managing companies' processes. They provide direction, enhance information sharing, and help solve conflicts.
- Coding and testing are crucial activities that fall in the system development life cycle (SDLC).
- Basis Path Testing is a white box testing technique that focuses on testing all paths in the program. It assists in defining all the execution paths and guaranteeing that they are run at least once.
- Cyclomatic complexity is a measurement tool that is used to determine the level of complexity that surrounds a program. It points to the static value derived from the number of linearly independent paths that exist in a program's source code.
- Control structure testing, also known as white-box testing, specialises in controlling program flow. It requires the creation of test cases that cover all branches of a program or application.

2.7 Keywords

- **Problem analysis -** Problem analysis in computer science involves understanding and defining the problem that needs to be solved before designing a solution. This stage is crucial for successful software development and system design.
- **Cyclomatic Complexity—Cyclomatic Complexity is a metric used to measure a program's complexity. It indicates the number of linearly independent paths through a program's source code and** helps determine the number of test cases needed for thorough testing.

- **Flowcharts -** Flowcharts are powerful tools for visualising and understanding processes. They offer clarity, improve communication, and facilitate problem-solving.

2.8 Self-Assessment Questions

1. What are the key components and steps involved in problem analysis?
2. Explain Algorithm Design.
3. What are the Components of a Flowchart?
4. What are the Key Concepts in Control Structure Testing?
5. What are the different Software Testing Techniques?

2.9 References / Reference Reading

- Balagurusamy, E. "Programming with Java: A Primer." 6th ed., McGraw Hill Education, 2019.
- Dasgupta, Sanjoy, Christos Papadimitriou, and Umesh Vazirani. "Algorithms." McGraw Hill Education, 2006.
- Dromey, R. G. "How to Solve It by Computer." Pearson Education India, 2006.
- Jaiswal, Aditya and Rajesh Kumar Shukla. "Algorithm Design: An Introduction." Wiley India, 2020.
- Srivastava, Amit, and Srivastava, Sumit. "Data Structures and Algorithms." BPB Publications, 2021.

Unit 3: Introduction to Programming Languages

Learning Outcomes:

- Students will be able to list different types of programming languages.
- Students will be able to describe the functions of language translators.
- Students will be able to compare machine language, assembly language, and high-level languages.
- Students will be able to interpret code snippets written in scripting languages.
- Students will be able to evaluate the advantages and disadvantages of using different

Structure:

3.1 Types of Programming Languages

Programming languages can be categorised into several types based on their characteristics, design philosophies, and intended use. Here are some common types of programming languages:

1. Low-Level Languages:

a. Machine Language:

- Description: Consists of binary code understood directly by the computer's hardware.
- Characteristics: Written in 0s and 1s, specific to the computer architecture, difficult for humans to read and write.

b. Assembly Language:

- Description: Uses mnemonic codes to represent machine instructions, providing a more readable format than machine language.
- Characteristics: It translates directly to machine code and is platform-specific, offering low-level control over hardware.

2. High-Level Languages:

a. Procedural Languages:

- Description: Focuses on procedures or routines that perform specific tasks.
- Examples: C, Pascal, Fortran.

b. Object-Oriented Languages:

- Description: Organises code into objects that contain data and methods to manipulate that data.
- Examples: Java, C++, Python.

c. Scripting Languages:

- Description: Used for scripting tasks, automation, and rapid development.
- Examples: JavaScript, Python, Ruby.

3. Declarative Languages:

a. Markup Languages:

- Description: Used to annotate text with structural and semantic information.
- Examples: HTML, XML, Markdown.

b. Query Languages:

- Description: Often used for structuring and retrieving information from a database.
- Examples: SQL (Structured Query Language).

3.1.1 Machine Language

Machine language, also known as machine code, is the lowest level of programming language that the computer understands in terms of the Central Processing Unit (CPU). It consists of a sequence of 0s and 1s binary digits that correspond to precise instructions aimed at the CPU. Every binary sequence relates to a specific operation or activity that the CPU performs, which puts machine language at the lowest level of programming.

Characteristics of Machine Language:

1. Binary Representation: Both instructions and data are binary values, which have only two states: 0 and 1.

2. Hardware Specific: Machine language instructions are closest to the computer's CPU architecture and instruction set. Machine language programs cannot be easily transferred to other computer platforms or operating systems.
3. Low-Level: Machine language gives users direct access to the computer architecture by allowing manipulation of memory, registers, and input/output devices.
4. Difficult for Humans to Read and Write: Machine language instructions are not easily readable by humans because they are in binary form; hence, they are not easily writable and can be directly understood by the computer.

Example of Machine Language Instruction: Example of Machine Language Instruction:

01001011 00000001

In this example:

- `01001011` may contain a binary opcode for the identified operation (for example, loading data from memory).
- `00000001` is the binary operand which defines the address in the memory from where the data is to be fetched.

Usage of Machine Language:

1. Bootstrapping: The computer's firmware or bootloader utilises machine language to start the hardware and download other higher-level software.
2. Operating Systems: An operating system's low-level components are typically written in machine language to communicate directly with the hardware.
3. Device Drivers: Device drivers that help in communication between the operating system and the hardware devices may include machine language code for efficiently handling crucial operations.

3.1.2 Assembly Language

Assembly is a group of instructions for a computer that is close to machine code and, at the same time, more understandable for a human. It incorporates alphabets and numbers to symbolise the instructions of the machine and locations in memory space, making it easier for the programmer to work with them. Being at the middle level of the software hierarchy, assembly language allows programmers to have fine-grained control over the computer hardware while at the same time limiting direct access to them.

Characteristics of Assembly Language:

1. Mnemonic Representation: Assembly language instructions are provided in words instead of numbers. For instance, instead of writing 000101, an assembly language will use the mnemonic MOV for the move, and instead of 001001, it will use the mnemonic ADD for addition.
2. Symbolic Names: The information stored in memory can be named (i.e., variables and labels) instead of using plain addresses, making the code easier to read and modify.
3. One-to-One Mapping: The instructions in assembly language are often one-to-one with the instructions in the machine code, making it easy to translate from one to the other.
4. Low-Level Control: Programs compiled in assembly language allow easy manipulation of hardware resources, including memory, registers, and I/O devices, hence providing a level of resource optimisation.

Example of Assembly Language Instruction: Example of Assembly Language Instruction:

assembly

```
MOV AX, 5
```

In this example:

- “MOV” is the abbreviation in code form for the move instruction.
- `AX` is a register to where the data is moved.
- `5` is a direct value moved into the `AX` register.

Usage of Assembly Language:

1. System Programming: Due to the fine-grained control it provides, assembly language is often used in system programming, writing device drivers, operating system kernels, and BIOS firmware.
2. Embedded Systems: Assembly language is used more frequently in embedded systems applications, where size and efficiency are paramount.
3. Performance-Critical Applications: Video games, graphics rendering, and real-time signal processing applications may include assembly language since certain performance-critical applications require low-level optimisation.

3.1.3 High-Level Language

High-level languages are relatively easy to write, read, and maintain and are easily understandable by anyone compared to low-level languages like assembly and machine language. They offer simple interfaces that prevent the software developer from worrying about the underlying hardware and come with features and constructs that can be used in software development. High-level languages enable the programmer to write a program without concern with the specifics of the low-level format.

Characteristics of High-Level Languages:

1. Abstraction: High-level languages are more abstract than low-level languages and enable the programmer to write algorithms and logic in a manner that is easier to understand.
2. Readable Syntax: High-level languages are similar to English in syntax and usage of meaningful words; hence, coded form is more understandable.
3. Portability: High-level language programs are generally platform and architecture-independent because compilers or interpreters are available to translate them.
4. Rich Standard Libraries: High-level languages also have large standard libraries and frameworks that contain many pre-implemented solutions to commonly encountered problems.
5. Automatic Memory Management: High-level languages include built-in tools like garbage collectors, which help manage memories and allocate and de-allocate them.

Examples of High-Level Languages:

1. Python: Popular for its easy coding and understanding; it is used in web development, data analysis, artificial intelligence, and automation.
2. Java: A resourceful language for developing desktop, web, and mobile applications as well as high-end business software applications.
3. C++ provides both system- and application-level features and is used in system programming, game development, and high-performance computing.
4. JavaScript: Primarily used in web design, JavaScript works both on the client and server-side and helps create web applications.
5. C#: Created by Microsoft, C# is widely used for assembling Windows applications, games, and business software using the .NET framework.

- **Knowledge Check 1**

  Fill in the Blanks

  1. __________ is also called native or machine code. (Assembly language)
  2. The lowest-level programming language is called __________. (machine language)

3. As a result of this, high-level programming languages that are intended for use by humans are________. (abstract)
4. Interpreters are very instrumental in the ___________ process. (execution)

- **Outcome-Based Activity 1**

  Make a PowerPoint presentation on High-level programming languages & Low-level programming languages

3.2 Language Translators

Compiler design involves language translators, which are important parts of the compiler that are used to translate source code in a high-level language to a form that is either machine code or equivalent. There are three main types of language translators used in the compilation process: There are three main types of language translators used in the compilation process:

Compiler:

- Description: A whole program in a high-level language is translated into either object code or byte code by a compiler.
- Process: Compiler passes through the lexical pass, syntactic pass, semantic pass, optimisation pass, and code generation pass.
- Output: It generates executable machine code or intermediate code that can run directly on the target platform or is further compiled by an interpreter or run time system.

Interpreter:

- Description: An interpreter compiles and executes the source code one line at a time or one statement at a time without forming the object file.
- Process: It provides a line-by-line analysis of the source code; it understands the line of code and performs the corresponding operations at that very instance.

Assembler:

- Description: An assembler translates assembly language into a machine language.
- Process: It translates mnemonic instructions and symbols into the binary form of those instructions and symbols.
- Output: The output is object files, including machine code instructions and symbolic addresses.

Translation Process:

1. Lexical Analysis:

- Divide the source code into tokens (such as keywords, identifiers, operators, etc.), excluding white space and comments.

2. Syntax Analysis:

- Ensures that tokens generated from the source code conform to the proper syntax of the language by parsing them according to the grammar rules.

3. Semantic Analysis:

- Finds semantic mistakes and ensures compliance with rules particular to certain languages (e.g., type checking or scope resolution).

4. Code Optimisation:

- Contains different procedures that improve the code and the resulting binary code (e.g., loop optimisation, constant folding).

5. Code Generation:

- Translates the language in which the source code is written into machine code or other code which can be executed directly on the target hardware or a virtual machine.

3.2.1 Compilers

- Translation Process: Compilers translate entire source code files from high-level languages like C, C++, Java, etc., into machine code or bytecode for the target platform/CPU. This process is normally done in stages such as lexical analysis, syntax analysis, semantic analysis, optimisation, and code generation.
- Optimisation: Compilers often pursue optimisations in a bid to improve the efficiency of the machine code produced. Optimisations may include removing duplicated code, rearranging instructions so that they run faster, or utilising specific processor characteristics to make the speed faster and the use of memory more efficient.
- Output: A compiler's result usually includes an executable file or object code, which can be run directly on the computer’s CPU without the need for the source code.

3.2.2 Assemblers

- Low-Level Language Translation: Assemblers are intended to convert an assembly language code into machine code. Assembly language is the symbolic language that uses assembler language to represent machine code instructions, and a mnemonic language represents each instruction.
- Direct Translation: Unlike compilers that translate source code in a high-level language to the intermediate forms, assemblers directly translate the assembly code

into the machine code. As for assembly language, each of the instructions in assembly language exactly matches a particular machine instruction.

- Output: An assembler usually generates an object file or an executable, which includes machine code that the CPU can run.

3.2.3 Interpreters

- Line-by-Line Execution: Interpreters run programs immediately while translating the source code into machine code that does not require compilation line by line. They look at each line of code and understand what it means before running the code simultaneously.
- Dynamic Typing: Interpreted languages may feature dynamic typing, which means that variable type may be changed during the program's execution. In contrast, with this flexibility, it becomes easy to program, but if it is not well managed, one becomes a victim of runtime errors.
- Portability: Interpreters are usually more portable than compiled programs that the interpreter directly runs. The interpreter itself must be installed on the target platform, but the same source code can be run on any platform supported by the interpreter.
- Examples: Interpreted languages are widely used, and languages such as Python, JavaScript, and Ruby are commonly used.

3.2.4 Editors

- Text Manipulation: Text editors are application software that allows users to create, modify, and manage text documents, including source codes.
- Features: Current text editors provide numerous functionalities which are specific to programming, like syntax highlighting, which colours the code to make it easier to read; intelligent suggestion, which includes code snippets or variable names; and code collapsing, which hides or shows the sections of the code to work on them separately.
- Integration: Some editors are interfaced with compilers, interpreters, or version control tools, through which developers can compile and test their code within the editor and efficiently control the code's versions.
- Customisation: Most text editors are very flexible. For example, one can download additional extensions or adjust the current parameters according to one's needs.

- **Knowledge Check 2**

Fill in the Blanks

1. Compilers directly convert entire source code files of a high-level language into __________. (machine code)
2. ____________ are intended to convert Assembly language to Machine language. (Assemblers)
3. ___________ run source code directly, with the help of which the code is executed line by line without compilation. (Interpreters)
4. Text editors offer ___________and ___________. (syntax highlighting and auto-completion)

- **Outcome-Based Activity 2**

  Compare Assemblers & Interpreters in class.

3.3 Introduction to Scripting Languages

These programming languages are particularly developed to facilitate scripting, otherwise known as automating. They include scripts that are usually interpreted, not compiled, which implies that an interpreter runs the code and does not need to be compiled first. Here's an introduction to scripting languages:

1. **Interpreted Execution:** Scripting language is interpreted, and the interpreter reads the source code line by line and executes the code. This enables one to develop and test quickly because modifications to the code can be run without going through the process of compiling it.
2. **Dynamic Typing:** Dynamic typing is popular among most scripting languages. It determines the type of a variable while the script is running. This makes coding easier, but if not handled well, it can result in run-time errors.
3. **High-Level Abstractions:** Scripting languages usually have more simplified forms of operations. They can have certain pre-included libraries containing different functions, such as file operations, string operations, and network operations. This makes them well-suited for automation, web development, and system administration-related jobs.
4. **Scripting vs. Programming:** Although scripting languages are used more frequently in small and specific applications, they can perform rather complex calculations while simultaneously being capable of handling large-scale applications. While the differences between scripting and programming languages

are slowly fading away, the extended features and capabilities of scripting languages are becoming more and more similar to the programming ones.

5. **Examples of Scripting Languages:** Some of the widely used scripting languages are Python, JavaScript, Perl, Ruby, and Shell scripting languages such as Bash. These languages are different, and each has its advantages and can be applied to specific tasks or fields.
6. **Scripting in Various Domains:** It is extensively used in web development using Javascript or Python with frameworks such as Django or Flask, system administration using Shell scripting, data analysis and machine learning using Python with libraries such as NumPy and TensorFlow, and automation using tools such as Selenium for web automation or Ansible for system configuration management.

3.4 Summary

- As for syntax, semantics, and their purpose, programming languages differ greatly from each other. Regardless of the type being low level or high level or procedural or functional, every kind of language is useful in some programming job and has its features and advantages.
- Machine language or machine code is the lowest level of programming language that can be executed directly by the CPU of a computer. It is a combination of linear binary digits, 0 and 1, that hold specific instructions for the CPU.
- Machine language can, therefore, be described as the basic mode of communication between the software and the hardware in a computer system.
- Assembly language is a type of low-level language that has a different representation of the instructions given to the computer. It incorporates alphabets and numbers to indicate the machine instructions and memory locations compared to machine language.
- These are languages that are written in clear form with detailed instructions that are easily understandable to the programmer and are relatively simpler to write and debug than low-level languages such as assembly language and machine language.
- Compiler design is distinguished by language translators who are called upon to translate source codes written in high-level programming languages into target

codes, which are essentially machine codes or other forms of equivalent representation.

3.5 Keywords

- **Machine language:** Machine language or machine code is the raw form of language that is directly understandable by the CPU of a computer.
- **Assembly language:** Assembly language is the closest language to machine code but in a format that is easier for a human to understand.
- **High-level languages for programming:** High-level languages are easier for humans to understand, write, and maintain than low-level languages such as assembly and machine code.

3.6 Self-Assessment Questions

1. What are Assemblers?
2. Explain Interpreters.
3. What do you mean by Machine language?
4. Describe Assembly language.
5. Justify High-level programming languages offer a balance between abstraction and efficiency.

3.7 References / Reference Reading

- Balagurusamy, E. Programming in ANSI C. 8th ed., McGraw Hill Education, 2021.
- Gottfried, Byron S. Programming with C. 3rd ed., McGraw Hill Education, 2019.
- Kanetkar, Yashavant. Let Us C. 18th ed., BPB Publications, 2020.
- Silberschatz, Abraham, Peter B. Galvin, and Greg Gagne. Operating System Concepts. 10th ed., Wiley India, 2018.
- Python Software Foundation. Python Programming: A Comprehensive Guide to the Language of the Future. BPB Publications, 2021.

Unit 4: Programming in C++

Learning Outcomes:

- Students will be able to define basic concepts of C++.
- Students will be able to explain the syntax and structure of C++ programs.
- Students will be able to demonstrate the use of variables and data types.
- Students will be able to apply control structures in C++ programs.
- Students will be able to develop programs using object-oriented programming concepts.

Structure:

4.1 Introduction to C++

C++ is a fast and efficient language that was created as an improved version of C. Developed by Bjarne Stroustrup in the early 1980s, it is based on the structure of C but adds object-oriented programming concepts. It is also used in most kinds of software development.

- **Key Features of C++**

1. Object-Oriented Programming (OOP):

C++ supports OOPS, a feature for developing objects and classes. This approach enhances modularity and code reusability by means of principles like inheritance, polymorphism, encapsulation, and abstraction.

2. Standard Template Library (STL):

The STL is a rather large library containing many template classes and functions. It offers structs known as vectors, lists, stacks, algorithms, and iterators, which make many programming operations easier.

3. Low-Level Manipulation:

As for C, C++ also gives direct access to the hardware and memory of a computer. This capability makes it perfect for use in system/software development for operating systems, game engines, and performance-oriented software.

4. Rich Library Support:

C++ also boasts extensive libraries that help in development, including raw codes for I/O operations, calculations, data manipulation, and many others.

5. Multi-Paradigm Support:

C++ supports multiple programming paradigms, including procedural, object-oriented, and generic paradigms. This means that a developer can use the most suitable strategy to address particular issues.

- **Basic Concepts in C++**

1. Variables and Data Types:

As it has been pointed out, C++ has many data types, such as primitive data types such as int, char, float, and double, and user-defined data types such as classes and enums.

2. Control Structures:

Control structures are control statements that determine the program's flow. They include `if`, `else`, `for`, `while`, and `switch`.

3. Functions:

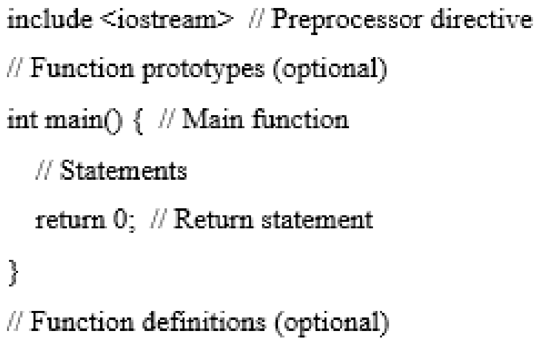
Functions in C++ make coding more manageable since it can be divided into manageable parts that can be called as many times as wanted. They can accept parameters and provide back a value.

4.2 Basic Syntax and Structure

some fundamental knowledge of C++ syntax and structures is central to the production of efficient and error-free programs. Here’s a breakdown of the fundamental components of C++ syntax and structure: Here’s a breakdown of the fundamental components of C++ syntax and structure:

1. Program Structure

A typical C++ program has the following structure:



```
include <iostream>  // Preprocessor directive
// Function prototypes (optional)
int main() {  // Main function
   // Statements
   return 0;  // Return statement
}
// Function definitions (optional)
```

- Preprocessor Directives: All lines that start with “ are preprocessor directives. They tell the compiler to precompile files or constants before the real compilation process begins.
- Main Function: The `main()` function is a starting or initial function for any C++ program, and it is already defined. Execution begins from the `main()` function.

2. Basic Syntax Elements

Comments:

- Single-line comment: `// This is a comment`

- Multi-line comment: `/* This is a
multi-line comment */`

Data Types:

- Some types of data are integers such as `int`, floating point numbers such as `float` and `double`, characters such as `char` and Boolean such as `bool`.

Variables:

- Variables are declared with a type followed by the variable name: `int age;`

Input/Output:

- This requires the use of '"include <iostream>"' for input and output operations.
- `std:`cout` is used to display (print) data to the terminal.
- `std: The `System. Console ().readLine()`:cin is used to read input from the console.

To effectively write and decode C++ programs, one must fully grasp the workings and semantics of the language. This includes understanding how to declare variables, control structures, functions, objects from classes, arrays, define functions, and call functions. These basics are important for preparing a good foundation for learning and further developing C++ programs.

- **Knowledge Check 1**

  Fill in the Blanks

  1. C++ is the extended version of the ___ programming language. (C)
  2. To include a standard input-output stream library in a C++ program, we use the directive ___ at the start of the code. (#include <iostream>)
  3. Every C++ program must have a main function, and that function must return an ___ type. (int)
  4. In C++, the statement `std::cout << "Hello, World!";` is used to ___ text to the console. (print)

- **Outcome-Based Activity 1**

  Write a C++ Program to print 'My First Program'.

4.3 Variables and Data Types

Variables in C++ are called the data locations in which data can be stored and accessed throughout the program. Every element in C++ is an instance of a variable that is characterised by the type of data it can include. Here's a comprehensive overview of variables in C++

1. Declaration and Initialisation

All variables must be defined before they can be used in the program. A declaration defines the variable, name and type and, where possible, the initial value of the variable.

Syntax:

type variableName;

type variableName = initialValue;

2. Basic Data Types

C++ provides several built-in data types: C++ provides several built-in data types:

- int: A type of variable that is used to store whole numbers and is of the integer data type.

```
int count = 100;
```

- float: Standard floating-point type for numbers with fractional parts; single precision.

```
float temperature = 36. 6;
```

- double: Float type for the floating-point numbers with more precision and Double type for more extended floating-point numbers.

```
double pi = 3. 14159;
```

- char: Appeal type for single characters.

```
char grade = 'A';
```

- bool: A type of Boolean that can only filter to be either ‘true’ or ‘false’.

```
bool isStudent = true;
```

- string: Available in the standard library for a sequence of characters.

```
include <string>
std::string name = "Bob";
```

3. Variable Scope

The scope of a variable determines where it can be accessed in the code: The scope of a variable determines where it can be accessed in the code:

- Local Variables: Program level variable that is defined within a function or a block and is only accessible within the function or block in which it is declared.

```
void myFunction() {
int localVar = 10; // localVar is localised to the myFunction.
}
```

- Global Variables: Determined outside all functions and can be accessed from any part of the program once they have been declared.

```
 Int global var = 20;  //, a global variable can be used in any function

 void myFunction() {
    globalVar = 30;  // Modify globalVar inside myFunction
 }
```

- **Static Variables:** Persist across several function calls of the control flow. Operates with the help of the `static` keyword that is used in the declaration of variables.

```
 void myFunction() {
     static int count = 0;  // count retains its value between calls
     count++;
     std::cout << count << std::endl;
 }
```

Data Variables are one of the key concepts of C++ programming and are used to store and work on the data. Knowledge about how to declare, initialize, and use variables is crucial for effective C++ programming; it is also vital to understand the variable's scope and constants. Having learned these basics, you will be able to manage data and regulate the activity of your C++ applications.

4.4 Control Structures (if-else, loops)

Conditional statements in C++ help to decide which part of the code should be executed next in a program. Some of these structures include decision-making statements, loops, and branches. Knowledge of control structures will enable one to write optimised and coherent code.

1. Decision-Making Statements

if Statement:

The `if` statement is used to run a statement or set of statements only when a certain condition is met.

Syntax:

```
if (condition) {
    // Code to execute if the condition is true
}
```

Example:

```
int age = 20;
if (age >= 18) {
    std::cout << "You are an adult." << std::endl;
}
```

If-else Statement:

The `if-else` type of statement enables one to perform one part of the code when a certain condition is true and another part of the code if the condition is false.

Syntax:

```
if (condition) {
   // Code to execute if the condition is true
} else {
   // Code to execute if the condition is false
}
```

Example:

```
int age = 16;
if (age >= 18) {
   std::cout << "You are an adult." << std::endl;
} else {
   std::cout << "You are a minor." << std::endl;
}
```

2. Loops

for Loop:

The `for` loop is a type of looping structure that allows a portion of code to be executed a predefined number of times.

Syntax:

```
for (initialisation; condition; update) {
   // Code to execute on each iteration
}
```

While Loop:

The `while` loop continues, repeating a set of instructions until the condition being tested is met.

Syntax:

```
while (condition) {
   // Code to execute as long as the condition is true
}
```

Do-while Loop:

The `do-while` loop is similar to the `while` loop, except it is executed at least once before the condition is checked.

Syntax:

```
do {
   // Code to execute
} while (condition);
``
```

4.5 Functions and Arrays

Functions in C++

Functions are basically reusable codes designed to carry out a particular task in a program. They assist in structuring code, improving its readability, and enabling reusable code.

Syntax:

```
returnType functionName(parameters) {
   // Function body
}
```

Key Concepts:

1. Function Declaration (Prototype): It states the function and its parameters, enabling the compiler to know how the function will be used even before its definition.

```
int add(int, int); // Function prototype
```

2. Function Definition: Holds the actual code which is to be executed in the given computer language.

```
int add(int a, int b) {
    return a + b;
}
```

3. Function Call: Calls for the other function to do what it is expected to do.

```
int result = add(5, 3);
```

4. Return Statement: This is employed to pass a value back from the function being called to the one that is called it.

```
return a + b;
```

Arrays in C++

The array is a list of elements of the same data type which are stored in adjacent memory locations. They enable you to place more than one value in one variable.

Syntax:

type arrayName[arraySize];

Initialisation: int numbers[5] = {1, 2, 3, 4, 5}; Because of this, it is said to be an example of array declaration and initialisation.

Accessing Elements:

The index, which begins at zero, is used to access the elements of an array.

```
int firstNumber = numbers[0] ; //get the first element
numbers[1] = 10; // This line change the second element
```

- **Knowledge Check 2**

  Fill in the Blanks

  1. A ___ is created to hold data that can be altered and accessed at different parts of a C++ program. (variable)
  2. The data type used to store integers in C++ is ___. (int)
  3. The data type which is used to store single characters in C++ is _____. (char)
  4. A ___ data type can contain a string of characters and is part of the C++ standard library. (std::string)

- **Outcome-Based Activity 2**

  Write a program for the addition of 2 numbers.

4.6 Object-Oriented Programming Concepts

Overview of object-oriented programming (OOP) concepts using C++:Overview of object-oriented programming (OOP) concepts using C++:

1. Classes and Objects: In C++, classes are used to define the form of how objects are to be created. A class defines attributes and methods that act on the attributes as well as objects that are an instance of that class. Objects are the actual occurrences of classes.

```
    class MyClass {
    private:
```

```
    int myData;
public:
    void setData(int data) {
        myData = data;
    }
    int getData() {
        return myData;
    }
};
```

2. Encapsulation: This is the packing of data with the functions that manipulate it into one unit known as a class. Data members are usually declared private, while member functions are declared public, which regulates data accessibility.

3. Inheritance: Inheritance is the process by which one of the classes can acquire properties and behaviours from another class. This enhances code reuse and makes it easy for one to create a class hierarchy in the application.

```
class DerivedClass: public BaseClass {
    // DerivedClass inherits members from BaseClass
};
```

4. Polymorphism: Polymorphism makes using objects of different classes but of the same superclass possible. This is done using the concept of function overloading, a case of compile-time polymorphism, and virtual functions, a case of runtime polymorphism.

```
class Shape {
public:
    virtual void draw() {
        // base implementation
    }
};
class Circle: public Shape {
public:
    void draw() override {
        // specific implementation for Circle
    }
};
```

5. Abstraction: This can be defined as concealing a class's internal workings while only making the class interface available to the outside world. Abstract classes have at least one pure virtual function – this makes them a template for the derived classes to follow.

```
class AbstractShape {
public:
   virtual void draw() = 0; // pure virtual function
};
```

6. Polymorphic Behavior: Actual types can be used to vary the behaviours of objects, which can help meet the requirements of flexibility and extensibility.

4.7 Summary

- C++ is an object-oriented, high-level programming language which was created as an extension of the C language.
- Along with procedural and object-oriented concepts, the STL and direct hardware access are among the most widely used languages for various purposes, including system and game software creation.
- Before proceeding into the details of C++, it is important to start by understanding the basics of the language and its syntax. This involves understanding how to declare a variable, use control structures, define functions and call them objects from classes, create arrays and manipulate them.
- Variables are important parts of C++ programming; they are used to store and manipulate data. Declaration, initialisation, and use of variables, as well as the scope of the variables and constants, are very important concepts when writing C++ programs.
- Control structures are very important in determining the direction or the flow of the C++ program. Knowledge of these constructs allows you to put a lot of complex thinking into a program and navigate its flow efficiently.
- Functions and arrays are basic concepts in C++ that increase your program's versatility and performance. Functions optimize your program and make it easier for you to reuse your code, while arrays store a set of data.

4.8 Keywords

- **Control structures:** The control structures play very vital roles in regulating the flow of a C++ program. Knowledge of these constructs makes it possible for you to perform sophisticated operations within a program and control the flow of the program.
- **C++:** is an advanced high-level language with high performance that has been created as an evolution of C language.
- **Variables:** Variables are one of the basic concepts in the C++ language, providing a means to store and use data.

4.9 Self-Assessment Questions

1. What are the Control structures?
2. Explain Functions and arrays are fundamental concepts in C++.
3. Justify C++ is a powerful, high-performance programming language.
4. Describe the OOPs concept.
5. Write a program to multiply 2 numbers. Take input from the user.

4.10 References / Reference Reading

- Balagurusamy, E. *Object-Oriented Programming with C++*. 8th ed., McGraw Hill Education, 2020.
- Kanetkar, Yashavant. *Let Us C++*. BPB Publications, 2020.
- Lafore, Robert. *Object-Oriented Programming in C++*. 4th ed., Pearson Education, 2018.
- Stroustrup, Bjarne. *Programming: Principles and Practice Using C++*. 2nd ed., Addison-Wesley Professional, 2014.
- Malik, D. S. *C++ Programming: From Problem Analysis to Program Design*. 8th ed., Cengage Learning, 2018.

Unit 5: Measures, Metrics, and Indicators in Software Engineering

Learning Outcomes:

- Students will be able to define key measures and metrics in software engineering.
- Students will be able to explain why metrics are important in software projects.
- Students will be able to identify common software metrics used in the industry.
- Students will be able to apply indicators to evaluate software quality.
- Students will be able to assess the effectiveness of software quality assurance processes.

Structure:

5.1 Importance of Measures and Metrics

Software measurement focuses on defining the size, quantity or dimension of a specific attribute of a software product or a process. This practice is important in software engineering to determine the software's quality.

ISO is the presiding body in determining the software measurement process; hence, the resulting measurements are standard across the industry.

Software Measurement Principles

The process of software measurement includes five key activities:

1. Formulation: Determining the right measures and assessment model for the software in consideration.
2. Collection: Collecting all the necessary information to compute the defined measures.
3. Analysis: Using formalism to work out the results and mathematical methods on the data.
4. Interpretation: Analysing the metric results to obtain information about the quality of the developed software.
5. Feedback: Make recommendations for the metric interpretations to the software development team.

Need for Software Measurement

Software measurement is necessary to:

- Determine the current quality of the software product or the process involved in developing the product.
- Make predictions of future qualities of the product or process.
- Improve the quality of the developed software.
- Supervise the project status on the aspects of budget and time.
- Backing of project planning and control by an efficient system of data collection and analysis.
- Determine problem areas and opportunities for optimisation.
- Adhere to the best practices of the industry and all the legal requirements of the country.
- Offer means of quantifying the characteristics of software products and the quality of the software development processes.
- Facilitate the making of the best practices of software development a continuous process.

Classification of Software Measurement

Software measurements can be classified into two types:

1. Direct Measurement: Gaining a quantifiable value by means of a standard scale on the product, process or attribute being studied.
2. Indirect Measurement: Estimating other related parameters to estimate the amount or degree of the attribute.

Software Metrics

Metric is defined as a measurement of the level to which any attribute relates to a system product or process.

Metrics are measurements that have been assigned to an attribute of a software product or process. Software metrics provide a quantifiable assessment of software attributes and serve four main functions: Software metrics provide a measurable evaluation of software attributes and serve four main functions:

1. Planning
2. Organising
3. Controlling
4. Improving

Characteristics of software Metrics

Effective software metrics should be:

- Quantitative: Capable of being quantified or having quantitative measures.
- Understandable: It is easy to compute, and there are clear definitions for it.
- Applicable: This may be applied right from the start of the first stages of creating software.
- Repeatable: The above results are consistent when the measurements are repeated on the same subjects.
- Economical: Economical in computation.
- Language Independent: It contains no coding and is not dependent on the use of any particular programming language.

In the context of software engineering, measures and metrics are used to evaluate and control several characteristics of the software development processes, products and projects. Here's how they are utilised in this domain:

Here's how they are utilised in this domain:

1. Process Metrics:

- Defect Density: Measures the cumulative number of defects detected for a given software as a ratio of the size of the software product (for instance, the number of lines of code). It assists in determining the quality and the necessity of the code improvement.
- Lead Time: This determines the duration from the commencement of the software development task to its completion. It assists in determining the effectiveness of the processes and the areas that require more attention.
- Cycle Time: Records the time a given phase of development, for example, the time between code commit and deployment takes to complete a cycle. They include help in the evaluation of agility and responsiveness.
- Velocity: This is an estimation of the quantity of work done by the development team in a specific cycle or sprint in agile development processes such as scrum. It is used in planning and monitoring a project's progress.

2. Product Metrics:

- Code Coverage: Measures the ratio of the program code to the extent to which test scripts or automated tests cover it. It assists in evaluating the performance of test suites and pinpointing the parts of the source code that are not well-tested.
- Maintainability Index: Estimates the accessibility of software changes and improvements by calculating characteristics such as the degree of code difficulty, code redundancy, and code volatility. It helps evaluate the code's feasibility over a long period.
- Performance Metrics: Calculate the real-time behavior of software, including response time, number of transactions per unit time, and consumption of system resources. They assist in making sure that the software meets the performance standards and pinpoint aspects that require improvement.
- Reliability Metrics: Determine the rate of software failure and its impact, i.e., the mean time between failure (MTBF) and mean time to repair (MTTR). These metrics aid in determining the software's stability and how resistant or immune it is to the above-mentioned factors.

3. Project Metrics:

- Budget Variance: Calculates the variation between the amount of money that was expected to be spent on a project and the amount of money that was paid. It is useful in monitoring the project's financial status and recognising when and where the costs exceed the set limits or how to minimize costs.
- Schedule Variance: This determines the extent to which the planned project schedules have deviated from the actual ones. It aids in measuring how close or far from the target a project is regarding the schedule and whether there are any slippages or complications.
- Resource Utilisation: Assesses the extent to which resources, such as developers' performance, server space, or tools, are effectively managed in a project. This assists in managing resources and increases project implementation's effectiveness.

Importance of Measures and Metrics

Measures and metrics play a crucial role in software engineering for several reasons:

1. **Performance Evaluation:** It depicts the performance of software projects, processes, and products by quantifying the outcome. This assists in evaluating areas for improvement and how resources can be best used.
2. **Quality Assurance:** Measures assist in evaluating the quality of software by evaluating things like the code's complexity level, the density of defects, and the coverage of tests. This helps teams remain professional and produce a good, dependable product at the end of the project.
3. **Risk Management:** By having such measures as effort, schedule, and budget indicators related to project progress, the stakeholders can detect the threats and work to respond to them as necessary.
4. **Decision Making:** Measures and metrics help obtain factual information for decision-making throughout the system's development cycles. They assist in functional decision-making and resource prioritisation to ensure that all strategic decisions are proper.
5. **Process Improvement:** Measures help constantly enhance the flow by pointing out the pinches in the software-creation process. This makes it possible for teams to effect specific changes and, hence, increase efficiency and productivity in the organisation.

Measures and metrics in software engineering are tools for controlling, analysing and assessing different sides of software development, which contributes to better results and customer satisfaction.

- **Knowledge Check 1**

  Fill in the blanks

  1. A ___________ is a characteristic of the size, quantity, amount or dimension of a certain feature of a product or a process. (metric)
  2. The software measurement process is described and regulated by __________. (standards)
  3. It is seen that the software measurement process can be characterised by ____________ & _____________ activities. (quantitative & qualitative)
  4. In ____________ measurement, the product, process, or thing is measured exactly with the help of a specific scale. (Direct)

- **Outcome-Based Activity 1**

  Make a presentation on Measures and Metrics and discuss them in the classroom.

5.2 Common Software Metrics

Measures of software productivity are numerous and include any measure that relates to the various aspects of software development, such as the quality of the code produced, the efficiency of the process used, and the project and team performance. Here are some of the most commonly used software metrics:

1. Lines of Code (LOC): Estimates the size of the codebase by getting the number of lines in the code. Though it is simple, it gives a basic idea of code size and, hence, the relative complexity and maintainability.
2. Cyclomatic Complexity: Measures a code's complexity level through the number of linearly independent paths associated with a program’s source code. Higher complexity values mean that code is complex and could be harder to test as well as maintain.
3. Code Coverage: Calculates how much more of a program or application can be tested with the same amount of effort and resources. It is used to evaluate the quality of testing and to determine parts of the code that are not tested enough.

4. Defect Density: This determines the rate at which defects or bugs occur per unit of code, which is measured in KLOC or a thousand lines of code. It gives an idea about the code's standard and where to focus on fixing bugs.
5. Lead Time: Determines the duration from the time work begins on a feature or a fix until the job is done. It allows for evaluating the effectiveness of development activities and finding ways to minimise time to market.
6. Cycle Time: Refers to the amount of time it costs to deliver one or more of the work items, for instance, a user story or a task. Cycle time is tracked to see where there are issues with the flow of work.
7. Velocity: Defines the speed at which a team delivers value by delivering the assigned user stories or features within a sprint or iteration. It is used to predict possible project schedules and evaluate the workers' efficiency.
8. Customer Satisfaction (CSAT): Obtains views from or on behalf of customers or other consumers to assess their level of satisfaction with the software product or service. CSAT scores are useful in understanding the user's experience and determining areas that need improvement.

5.3 Indicators in Software Engineering

Measures in software engineering are quantitative parameters that give an idea of different facets of the software development process, the product, or the team. Here are some important indicators: Here are some important indicators:

1. Efficiency Indicators:

- Lead Time is the time it takes to complete a feature from its initial initiation to the delivery time.
- Cycle Time: This is the amount of time it takes to complete one cycle of work, that is, a single phase of work.
- Throughput: Quantifies how many work items are delivered in a given period.

2. Quality Indicators:

- Defect Density: Expresses the number of defects or bugs in terms of the lines of code.
- Code Coverage: Estimates the code coverage ratio, which signifies the extent to which the code has been tested.
- Code Review Effectiveness: Quantifies the amount and quality of work done with regard to code review.

3. Process Indicators:

- Agile Metrics (Velocity, Burndown/Burnup Charts): Plan and monitor work progress using Agile software development methods.
- Work in Progress (WIP): This refers to the quantity of work processes or functions that are being implemented simultaneously.
- Sprint Retrospective Actions Completed: It keeps track of an item that needs to be accomplished following a sprint retrospective.

4. Team Performance Indicators:

- Velocity: Estimates the speed with which value is created in a team by the number of user stories or features accomplished.
- Team Satisfaction: Solicits views from its members to determine the team's satisfaction level with the organisational climate and the workflow.
- Employee Turnover Rate: This calculator calculates the turnover rate for team members, which implies the rate at which team members are leaving the organisation and may, therefore, affect the team's dynamics.

5. Customer Satisfaction Indicators:

- Net Promoter Score (NPS): This assesses the potential of the customer to endorse the product or service to others.
- Customer Support Response Time: Measures the time taken to manage complaints/queries or the time taken to find a solution to a particular problem.
- Customer Feedback Ratings: This process gathers information from customers to measure their satisfaction with the developed software product or service.

6. Risk Indicators:

- Requirements Volatility: Assesses how often and to what degree requirements change in a project.
- Technical Debt measures how much maintenance is delayed or how much of the code was designed suboptimally.

- **Knowledge Check 2**

  Fill in the Blanks

  1. ____________ Proportionizes the degree of the program’s entangled code by counting the number of linearly independent paths through the program’s source code. (Cyclomatic complexity)

2. ___________ offers additional fundamental information about the quality of the codebase and is useful for choosing which bugs to fix first. (Code coverage)
3. CSAT stands for ____________. (Customer Satisfaction)
4. _____________ Records the time that is taken to resolve customer complaints and the time it takes to attend to customer queries. (Response time)

- **Outcome-Based Activity 2**

  Do a discussion on Quality Indicators & Efficiency Indicators in the classroom.

5.4 Software Quality Assurance

Software Quality Assurance (SQA) can be referred to as involving the systematic management and control of software products and processes to meet specified requirements and standards from the time of their conception to when they are delivered. Here's an overview:

1. Goal of SQA: The main purpose of SQA is to ensure that the software products developed are correct, adequate, consistent, reliable, and delivered as planned for and within the required time and costs.

2. Key Activities:

- Quality Planning: Creating a roadmap that details the quality assurance activities, procedures, and measures to be adopted throughout the project.
- Quality Control: Keeping track of the software development processes to check for compliance with the quality standards and identify defects or lack of compliance with the requirements.
- Testing: Unit testing, integration testing, system testing, and acceptance testing to ensure the software is right and working as it should.
- Reviews and Inspections: Code and design reviews and inspections in order to detect faults and other defects at the early stage of software development.
- Process Improvement: Repeating the improvement of the developed processes for software creation based on the feedback, measurements, and information received from previous projects.

3. Principles and Standards: SQA principles include the following;

- SQA is transparent,
- SQA is accountable,
- SQA is repeatable,

- SQA is improving.

It also follows the standards and guidelines of its kind, including ISO/IEC 25010 on software quality models and ISO/IEC 12207 on software lifecycle processes.

4. Tools and Techniques: SQA also uses many tools to aid in quality assurance processes, such as version control tools, testing tools, code review tools, and bug tracking systems.

5. Roles and Responsibilities: SQA requires the involvement of various people in a project, such as software developers, testers, managers, and quality assurance specialists. Both roles have specific responsibilities regarding the quality of the software at each stage of development.

6. Benefits of SQA: The literature demonstrates that SQA has an overall impact on better software quality, lower software development costs, happier customers, and more reliable and manageable software products.

5.5 Summary

- A measurement can be defined as a statement of a specific size, quantity, amount or dimension of a procedure or a product.
- In software engineering, measures and metrics are used to determine and enhance the characteristics of software development processes, products, and projects.
- Measures and metrics are essential in software engineering for the following reasons: Performance Evaluation, Quality Assurance, and Risk Management, among others.
- Software metrics refer to a broad range of measures that cover all aspects of the software development process, such as code quality, workflow, project planning, and individuals' activity levels.
- Some of the most commonly used software metrics are measures of software quantity: Lines of Code (LOC), Cyclomatic Complexity, Code Coverage, etc.
- Measures in software engineering are quantitative variables that can be used to monitor different aspects of the processes, the product, and the performance of a team involved in the software development process.
- There are principles of Software Measurement such as Formulation, Collection, Interpretation and so on.

5.6 Keywords

- **Measurement:** A measurement is a statement of a specific size, quantity, extent, or calibre of a process or good.
- **Metrics:** Metrics in software engineering are quantitative parameters that offer information regarding one or more of the aspects of the software development process, the product, and the team.
- **Net Promoter Score (NPS):** This is the level to which a customer will refer others to use a particular product or patronise a certain service.

5.7 Self-Assessment Questions

1. What are the Importance of Measures?
2. Discuss Software Measurement Principles.
3. What is the Importance of Metrics?
4. What are the commonly used software metrics?
5. Explain Indicators in Software Engineering.

5.8 References / Reference Reading

- Pressman, Roger S., and Bruce R. Maxim. Software Engineering: A Practitioner's Approach. 9th ed., McGraw-Hill Education, 2019.
- Sommerville, Ian. Software Engineering. 10th ed., Pearson Education, 2015.
- Jalote, Pankaj. An Integrated Approach to Software Engineering. 3rd ed., Springer, 2005.
- Desikan, Srinivasan, and Gopalaswamy Ramesh. Software Testing: Principles and Practices. Pearson Education India, 2007.

Unit 6: User Interface Design in Software Engineering

Learning Outcomes:

- Students will be able to define the principles of user interface design.
- Students will be able to explain the concept of user-centred design.
- Students will be able to create prototypes for user interfaces.
- Students will be able to apply usability testing methods.

Structure:

6.1 Principles of User Interface Design

User interface design in software engineering, also known as Usability, is a significant sub-discipline that deals with designing interfaces that enable interaction between users and an application. The main objectives are centred on the concepts of form, utilitarianism, and simplicity.

Here are the key aspects of UI design in software engineering:

1. Clarity: Interfaces should communicate information and functionality to users. This means the simple text, icons, and any other nodes and links must be easily understandable.
2. Consistency: The aspects of design are made uniform across the application to improve the learnability of patterns and expectations. The consistency of colours, fonts, and layouts enables the users to easily find what they are looking for or perform certain tasks in the software.
3. Feedback: It will be appropriate if users are able to receive feedback for their actions as soon as they carry them out. This could be in the form of visual stimuli (as in underlining a button once it is clicked), audio feedback, or any other form of positive feedback that acknowledges the user's action.
4. Efficiency: The users should be able to accomplish tasks directly and easily using the interface. Measures that can do this include avoiding time-consuming shortcuts, clear and easily understandable menus, and unnecessary sub-steps in any process.
5. Aesthetics: A good interface design also results in user satisfaction and engagement. This includes the right proportions, choice of colours, and sharp and clear pictures or graphics.

UI Design Process

1. Requirement Gathering: By interviewing the target users, administering questionnaires or conducting user-based research.
2. User Research and Personas: Based on actual research about users, one can create user personas and then easily design the interface to suit each user category.
3. Wireframing and Prototyping: Wireframes and prototypes are used to design the simplest view of the layout and organisation of the interface, and interaction design is done using prototypes. Some of the applications used in this aspect include Sketch, Figma, and Adobe XD.

4. Design and Development: Designing the wireframes with the help of the visual design chosen and implementing the interactivity. This is typically a process of the UI designers, UX designers, and developers.
5. Usability Testing: This is the process of carrying out tests with actual product users to determine their usability problems. This assists in modifying the interface based on users' responses.

Principles of UI design

User Interface (UI) design principles are an important set of principles that guide the development of interfaces that look good and are effective and easy to use. Here are the core principles of UI design: Here are the core principles of UI design:

1. Clarity

The main objective of the UI design is to ensure that interfaces are comprehensible. Users should get a clear idea of what the interface is about and how they can operate it. This includes:

- Clear Visual Hierarchy: Structuring items to create a visual flow that takes the user's glance towards the desired areas of the interface.
- Legible Text: Reading fonts, font sizes, and non-clashing colours with the text and the background or the theme of the document.

2. Consistency

Consistency means that the users will know and be able to anticipate the way the interface will function. This principle can be applied to: This principle can be used to:

- Design Elements: Maintaining the consistency of all the colours, fonts, and icons used in the application.
- Behaviour: Making sure that the same thing yields the same result anywhere in the whole interface.

3. Feedback

From our study, feedback can be defined as the information that users get concerning the results of their actions. Effective feedback mechanisms include:

- Visual Cues: This means that an element is highlighted or differs in colour or form when it comes to interaction (for instance, buttons' changes).
- Audio Signals: Alerts/sounds produced in response to activities (for instance, alert for an error).

4. Efficiency

Easy interfaces allow a person to complete a task with little effort. Key aspects include:

- Shortcuts: Offering access keys for the power users.
- Streamlined Workflows: ensuring that the client goes through the fewest steps to get to the result.

5. Accessibility

Here, an open interface means that all users, including the physically disabled, should be able to use the application. This involves:

- Text Alternatives: Describing images that are used in the content to make them accessible to impaired individuals.
- Keyboard Navigation: All features of a function are easily available on the keyboard.

6. Flexibility

Flexibility enables the users to access this interface in the manner that they prefer. This includes:

- Customizability: Enabling users to customise interface settings.
- Adaptability: To check that the interface is functional and looks good on different devices and with different resolutions.

7. Simplicity

Simplicity eliminates any clutter by targeting the reduction of unnecessary components and features. This can be achieved by: This can be achieved by:

- Minimalism: This means that simple designing is used because of the elimination of unnecessary icons and designs.
- Focused Content: Displaying the content and services that are useful for the users in the current context.

8. Error Prevention and Recovery

Avoiding errors as much as possible is the best policy, but should the worst happen, the user should be able to correct the mistake quickly. This principle includes:

- Preventive Measures: Making the interface safe (e.g., preventing the choice that is not available at the moment).
- Undo Options: Enabling users to cancel the operation they have done.

9. Feedback and Response Time

Another factor is the fast reaction of the system to the actions of a user. This includes:

- Immediate Response: It often provides real-time feedback on actions to prevent users from becoming frustrated with the products.

- Loading Indicators: Employing progress bars or spinners to inform the users that the system is working on the request.

- **Knowledge Check 1**

  Fill in the Blanks

  1. Clear _______ helps users understand the interface quickly. (layout)
  2. Some _______ elements act more 'predisposedly' to minimize the learning effect. (design)
  3. High contrast provides _______ for all the users. (clarity)

- **Outcome-Based Activity 1**

  Students need to apply the principles of User Interface (UI) design principles to create a user-friendly app mockup, focusing on clarity, consistency, feedback, efficiency, and accessibility.

6.2 User-Centered Design

User-centred design (UCD) is a model and process in software engineering that aims to understand and meet the end users' needs in the execution of their tasks. The objective is to develop products that are functional, effective, and pleasurable to the target consumer. Here's an overview of the key elements and process of UCD:

Key Elements of User-Centered Design

The following are the key elements of the user-centered design:

1. User Involvement: Some of the techniques that should be followed in the design include interacting with the users at every stage of the design process to get their feedback.
2. Empathy: The fundamental concept of user-centered design is the method of focusing on the aspects that matter to users, their issues, and their visions.
3. Iterative Design: Iterative improvement of the design based on user feedback and system testing.
4. Usability: Ways to make the product easy to use and learn and also give the users a satisfactory experience.

The User-Centered Design Process

1. **Research and Understanding**

- User Research: Carrying out research in order to identify the user, what they want, how they interact with the system and what frustrations the user has. Some of these are interviews, surveys and ethnographic research.
- Personas: Establishing detailed personas depicting major user groups to help make design choices.
- User Journeys: Defining all the activities a user goes through to get to the point where they will use the product and analysing the points in this process, which can be critical.

2. **Design and Prototyping**

- Requirements Gathering: Identifying what needs to be done with a product based on the findings from the user research.
- Wireframing: Constructing rough sketches of the interface where major divisions and placement of objects are defined but with no concern for appearance.
- Prototyping: Building the actual models of the product to test the model in a real-life situation. Some of the frequently used software are Figma, Sketch, Adobe XD, and others.

3. **Evaluation and Testing**

- Usability Testing: This involves using real users' interactions with the prototype to determine its usability problems and reception. It can take the form of moderated sessions, A/B testing, or any other remote testing tool.
- Iterative Refinement: Design modification, modification implementation, and testing phases where changes are made in the design to fit the results obtained from testing and then the testing is done again to make sure the new changes are effective.

4. **Implementation and Deployment**

- Development Collaboration: Coordinate with developers so that they can implement the design correctly and in a user-friendly way.
- Ongoing Feedback: The ability to systematically gather feedback from users once the application has been deployed to improve it further.

User-Centered Design and Its Application

- Early and Often User Involvement: Incorporation of the users at the initial stage of the design process and continuous interaction with them.
- Contextual Understanding: Developing with reference to the user's context, that is, the situation in which the product is to be used.

- Inclusive Design: This includes accommodating many users and the disabled so that the product can also meet their needs.
- Clear Communication: Making sure all relevant parties are well-informed of the design decisions and the reasons behind them by documentation and proper briefing.

6.3 Prototyping and Usability Testing

Prototyping

Prototyping in User Interface (UI) design within Software Engineering (SE) is an imperative stage that is used to ensure that there is a link between the theoretical and the actual. It involves the development of a first approximation of the user interface to assess the proper design, besides making sure that the final product will be efficient for use. Here’s a detailed overview of the role and process of prototyping in UI design within SE:

Importance of Prototyping in UI Design

1. Visualisation of Ideas: Prototyping makes it easy for designers to contextualize their clients' ideas by enabling them to identify the final product to be developed.
2. User Feedback: Users can be engaged in early prototypes created so that the proper decisions can be made in the design process.
3. Iterative Improvement: Prototypes can also be used in an iterative process in which designs are repeatedly modified according to users and stakeholders.
4. Cost-Effective: Preventive measures are cheaper than curative measures when it comes to dealing with problems regarding design requirements.
5. Enhanced Communication: Designers, developers, and stakeholders use them as a common reference point so that everyone is on the same page.

Types of Prototypes

1. Low-Fidelity Prototypes: These are very basic elements that may not even involve any interaction with the user; they are mainly concerned with organisation.

- Paper Prototypes: To get first impressions, lo-fi prototypes, such as sketching on paper or printed drawings of the screens, are used.

- Digital Wireframes: Low-fidelity wireframes made with applications such as Balsamiq or Moqups that position the objects in the frame without elaborating on their design.

2. High-Fidelity Prototypes: These are realistic, tangible models that are usually three-dimensional and possess all the characteristics of the actual product.

- Clickable Prototypes: Prototypes made with Figma, Sketch, Adobe XD, or InVision that allow the user to move through the design.
- Functional Prototypes: These include interactive mockups and live simulations of the final product, which can be developed using HTML/CSS/JavaScript or Framer, etc.

Prototyping in UI Design

1. Define Objectives and Scope

- Clarify the purpose of the prototype when developing a feature, designing prototypes, or presenting them to other interested parties.
- Establish the range to exclude elements that are not necessary while meeting all the crucial requirements.

2. Gather Requirements

- Gather user requirements from research techniques like interviews, questionnaires, and observations.
- It is crucial to develop user personas and scenarios that would help in design choices.

3. Initial Design and Wireframing

- Create mockups that show the primary structure and design of the interface.
- Mainly, pay attention to the three links and their general layout.

4. Build the Prototype

- To create the prototype, the right tools should be employed depending on the initial designs.
- Integrate recommendations from the other team members and stakeholders.

5. Testing and Iteration

- It is recommended that real users be used and usability testing performed to collect user feedback and check for usability problems.

- Modify the design based on the testing outcomes and adjust the prototype for better usability and stability.

6. Refinement and Finalisation

- Gradually transition from low-fidelity to high-fidelity; include further engraved features and limited functionality.
- Make sure that the final prototype includes all the features intended for the final product.

Usability testing

Usability testing is one of the most important stages of user-centred design in the process of software engineering. It involves the assessment of a product or a system based on the users' interaction with it. Its main purpose is to find any usability problems, collect quantitative and qualitative data, and define a user's satisfaction level with a certain product. Here's an in-depth look at usability testing: Here's an in-depth look at usability testing:

Importance of Usability Testing

1. Identify Issues Early: It is always easier and cheaper to identify problems in usability in the initial stages of design rather than when the product is almost complete.
2. Improve User Satisfaction: Making the product friendly to the user also ensures that the user is satisfied with the product and hence becomes sticky with the product.
3. Validate Design Decisions: Testing with users provides proof that some of the design decisions made were correct.
4. Enhance Product Quality: A product with fewer usability issues and a more satisfying user experience is usually more successful in the market.

Types of Usability Testing

1. Exploratory Testing: Typically carried out in the preliminary stage of the design process to understand the users' behaviour and their perception of certain design ideas.
2. Assessment Testing: This type of testing examines a particular product or sketch of a product to determine specific problems with usability.
3. Comparative Testing: Covers two or more designs to identify which one is superior in the aspect of usability.

The Usability Testing Process

1. Define Objectives and Goals

- Define the objectives of the testing, for example, if you are looking for specific usability problems or if you aim to confirm users' satisfaction.

2. Recruit Participants

- Recruit participants who are typical users of the targeted technologies.
- This is where different methods, such as screening questionnaires, must be applied to ensure that participants belong to the user personas.

3. Define Test Scenarios and Tasks

- It is necessary to create realistic scenarios that the users are most likely to come across in their day-to-day activities.
- Create scenarios that reveal functional activities that customers require to do with the product.

4. Conduct the Test

- In the case of moderated sessions, help participants perform the tasks and analyse their actions.
- When designing unmoderated sessions, it is recommended that clear guidelines be given and that the testing tool be designed to collect specific information.

5. Collect Data

- It is necessary to collect qualitative data concerning users' opinions and actions and quantitative data regarding task performance rates and time spent on tasks.
- Try to record the sessions where possible to go over them at a later time.

6. Analyze Results

- Identify trends, problems and the areas that users have difficulties with.
- Group issues according to their severity and the frequency of occurrence.

7. Report Findings

- Prepare a comprehensive paper to present major discoveries, conclusions, and proposed strategies.
- It is important to provide both qualitative data and quantitative results.

8. Iterate and Improve

- It is recommended that the findings be used to make design improvement decisions.
- Other additional rounds of testing should be conducted to further check the updates and confirm the constant enhancement.

- **Knowledge Check 2**

  Fill in the Blanks

1. Making a __________ version of the product assists in the process of visualising the looks as well as the requirements of the design as well as the performance of the functions before the final production. (Prototype)
2. Usability tests should be done using real users to enable one to discover ____________ issues and obtain user feedback that can be used to enhance the usability of the interface. (usability)
3. Such elements are simple and frequently non-annotated diagrams such as __________ or wireframes that highlight the layout and organisation. (mockups)
4. These detailed and intricate models are as near as can be to the final product in terms of design and ______. (functionality)

- **Outcome-Based Activity 2**

  Students need to create a digital or physical prototype of their app using prototyping tools or paper sketches.

6.4 Human-Computer Interaction

Human-computer interaction (HCI) is an interdisciplinary branch of science and engineering that deals with the design and use of computers and their unified environment in the production of systems to support human use. It describes the major characteristics and phenomena of such systems.

Here's an overview:

1. User-Centered Design:

The focus of HCI is on usability and user experience. When designing interactive systems, it involves considering the user, the actual and potential user activities, and the context. It tries to establish methods for making the intended systems effective, easy to use, and even enjoyable to the users.

2. Elements of HCI:

- Users: Knowledge regarding users' characteristics, such as their mental and physical health, cultural endowment, and inclinations.
- Tasks: Understanding the tasks and purposes that the users perform with the help of the interactive system and creating interfaces to provide effective solutions.

- Context of Use: The factors affecting people's interaction with the system include geographical location, time, and social context.
- Interfaces: Software that enables users to interact with computers through graphical interfaces, voice commands, touch screens, and other forms of interaction with the computer.
- Feedback and Response: Provide the user with feedback about the system's state and/or actions and respond appropriately and promptly to the user's commands.

3. HCI Process:

- Research: People may acquire data about the users, tasks, and contexts through interviews, questionnaires, and observation.
- Design: Designing and modelling interactive systems from the conceptual models, actual models, and mock-up models based on user specifications and feedback.
- Evaluation: Performing assessment and usability issues on the actual interactive systems with the actual users to improve them.
- Implementation: Designing and implementing the final prototype of the interactive system as suggested by the design and the evaluation outcomes.

4. HCI Principles:

- Usability: A measure of the degree to which the system and the tasks performed on it are easy to learn, fast to perform, and pleasurable to the users.
- Accessibility: Guaranteeing usability of the interactive systems for both the conventional and the disabled ability users.
- User Experience (UX): The feelings, impressions, attitudes, and level of satisfaction of users while they are using a particular system.
- Human Factors: Cognitive aspects that pertain to human features and constraints when it comes to the use of interactive systems like attention and memory, as well as motor coordination.

6.5 Summary

- In software engineering, User Interface design, also known as UI design, is a vital discipline that deals with designing interfaces that enhance user-software communications.

- Designers can develop and design good user interfaces by following a standard set of guidelines, adhering to the design process, and keeping up with current trends and technologies.
- User interface (UI) design best practices are standard rules that guide the design of interfaces with appealing graphics and more functional features.
- Prototyping in User Interface (UI) design within Software Engineering (SE) is an important stage that enables the implementation of a transition from a theoretical concept to the actual application. It involves producing mock-ups of the interface to be used in realising the intended designs for the created application and creating a final product that can be easily used.
- There are two categories of prototypes, namely Low-Fidelity Prototypes & High-Fidelity Prototypes.
- Using prototypes, in this case, is an iterative approach that makes it easier to achieve low development costs, less misunderstanding between designers and developers, and, in the long run, a better user experience.
- Usability testing is an important activity in the user-centred design paradigm in software engineering. It is a technique for assessing the quality of a product or a system with the help of actual users.

6.6 Keywords

- **User Interface:** User interface (UI) design in software engineering is a significant branch that strives to produce interfaces that enable users to interact with applications.
- **User-Centred Design:** User-centered design (UCD) is a framework and process in software engineering that emphasizes the end-users of systems and interfaces.
- **Prototyping is a topical phase of User Interface (UI) design in Software Engineering (SE), which aims to link** the conceptual phase with the materialisation of the concept.

6.7 Self-Assessment Questions

1. What are the principles of User Interface?
2. What do you understand about the term prototyping?
3. Explain Usability testing.

4. Discuss User-Centered Design in detail.
5. What are the types of Prototypes?

6.8 References / Reference Reading

- Johnson, Jeff. Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules. 3rd ed., Morgan Kaufmann, 2020.
- McKay, Everett N. UI is Communication: How to Design Intuitive, User-Centered Interfaces by Focusing on Effective Communication. Morgan Kaufmann, 2013.
- Wathan, Adam, and Steve Schoger. Refactoring UI: The Book. Refactoring UI, 2018.
- Shneiderman, Ben, and Catherine Plaisant. Designing the User Interface: Strategies for Effective Human-Computer Interaction. 6th ed., Pearson, 2017.
- Sinha, Piyush, and Sanjeev Sharma. User Interface Design and Evaluation: Theory and Practice. Pearson Education India, 2022.

Unit 7: Introduction to MS Word

Learning Outcomes:

- Students will be able to identify the basic functions of MS Word.

- Students will be able to explain how to create and save documents.
- Students will be able to demonstrate how to edit text in a document.
- Students will be able to apply formatting techniques to documents.
- Students will be able to utilize tools like spell-checking and mail merge in document creation.

Structure:

7.1 Creating Simple Documents

The beginning of Microsoft Word can be an interesting experience, especially for the new users of the software. Here's a step-by-step guide to help you get started with Microsoft Word and unleash your creativity:

1. **Launching Microsoft Word:** Microsoft Word begins with identifying the application icon on the desktop or in the Start menu and double-clicking. When you click on the program you created, a new document should appear, which initially is blank.

2. **Exploring the User Interface:** Take some time to get to know the application's design and features. The major parts are the Ribbon, which comprises several tabs with related commands; the Quick Access Toolbar, which is useful for quick access to some commands that you will be frequently using; and the workspace, the document area where one will be composing content.

3. **Creating a New Document:** To design a new document, go to the home tab in the ribbon and click on “new.” Depending on the type of document, you can start a blank document or use the templates provided, such as a resume, letter, or report.

4. **Adding and Formatting Text:** Begin writing your content in the writing zone, which is the document area. To format the text, highlight it, then click on the formatting tool in the ribbon at the top. To set the style and size of the font, make the text bold or italicised, align the text, and space out the paragraphs in the way you want.

5. **Inserting Images and Graphics:** To improve the appearance of the document, images, illustrations, or other graphics can be added. Select the desired option in the Insert tab of the Ribbon. Pictures can be imported from your computer or other sites, and illustrations in the form of shapes and diagrams can also be drawn.

6. **Formatting and Organising:** Microsoft Word offers various formatting options that give the document a good appearance. Headings, bullet points, and numbered lists enhance the structure of the information you are conveying to your audience. Tables can also be used to present information, especially when it is organised in tabular form.

7. **Saving and Managing Documents:** When typing work on a computer, it should be noted that you have to save your work from time to time. To save your document, go to the “File” tab, then choose “Save As” and a directory on your hard disk where you’d like to save the document. Label it descriptively, and select the right file format that would be recognisable by other applications (for instance, . docx).

7.2 Editing Text

Editing text in Microsoft Word is one of the key skills that any user of this popular word processor should possess. Here's a comprehensive guide on how to effectively edit text in MS Word:

- **Basic Text Editing**

Selecting Text

To make any changes, you first have to highlight the text that you want to modify. This can be done in several ways: This can be done in several ways:

- Click and Drag: Left click at the start of the text, and hold clicking until the mouse pointer reaches the end of the text.
- Double-Click: Double-click to select.
- Triple-Click: Select a whole paragraph by triple-clicking any of the clicking keys.
- Shift + Arrow Keys: While pressing the Shift key, use the arrow keys to expand the selection.

Deleting Text

- Backspace Key: Erases all the characters to the left of the cursor on the screen or the current cell, if there is one.
- Delete Key: This key erases the character to the right of the cursor on the active sheet or within a selected cell.
- Highlight and Delete: Highlight the text you want to delete, then delete the selected text using the backspace or delete button.

- **Advanced Editing Techniques**

Copy, Cut, and Paste

- Copy (Ctrl + C): Cuts the selected text to the clipboard.
- Cut (Ctrl + X): Cuts the selected text and places it on the clipboard for subsequent pasting into any application.
- Paste (Ctrl + V): Copies the text in the clipboard, then pastes it at the cursor's point of location.

Find and Replace

The 'Find and Replace' option is helpful for finding the typed text and replacing it with the new text.

- Find (Ctrl + F): Toggles the navigation pane in order to search for the text.
- Replace (Ctrl + H): Opens the dialogue box for replacement, where you can search for text to replace.

Formatting Text

Text formatting alters the style of the text. This includes:

- Font Style: It is located under the Home tab, and you can change the font type.
- Font Size: Used to increase or reduce the font size from the Home tab.
- Bold, Italic, Underline: Use these styles by clicking on the buttons on the Home tab or by hot keys (Ctrl + B, Ctrl + I, Ctrl + U).
- Font Colour: Alter the colour of the text to the font colour option located in the Home tab of the toolbar.

Paragraph Formatting

- Alignment: This can be done from the Home tab using the options align left, centre, right or justified.
- Line Spacing: Change the line space from the drop-down list available in the Paragraph group in the Home tab.
- Indentation: Use indents from the Paragraph group by either moving up or down.

7.3 Working with Tables and Graphics

Microsoft Word's tables and graphics functions make it possible to present data systematically and make documents more attractive. Here's a comprehensive guide on how to work with tables and graphics:

- **Tables:**

1. Inserting a Table:

- Put the mouse pointer at the point where you wish to add the table.
- Move to the "Insert" located at the top of the window, and click on "Table".
- Select how many rows and columns you want your table to have from the grid or click on Table and then Insert Table.

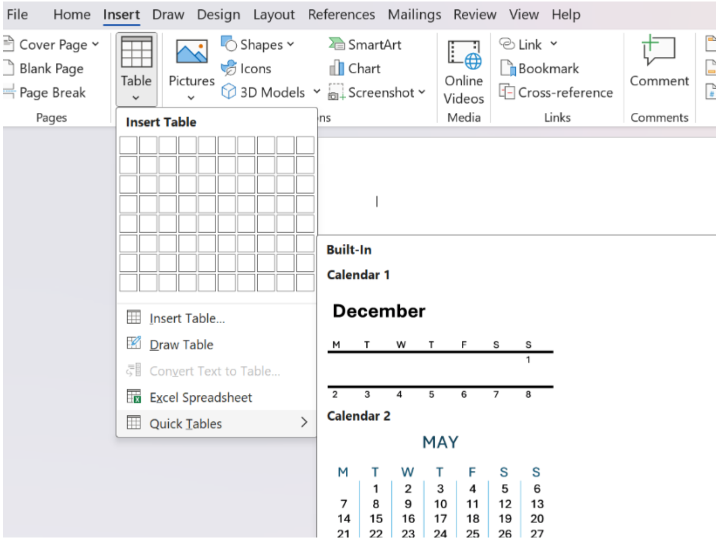


Fig: Inserting a Table

2. Formatting a Table:

- Click on the table you want to format; you can now format it from the Table Tools tabs in the ribbon, namely Design and Layout.
- Adjust the borders, shading, and alignment. Make the row and column sizes flexible according to the requirements.

3. Adding and Deleting Rows/Columns:

- To insert a row, move the cursor on the specific row that is next to the place you want to make a new row, then go to the Layout tab and click on “Insert Above” or “Insert Below.”
- To insert a new column, you need to position the cursor on the casing column next to the new column that is to be added and click either the “Insert Left or Insert Right” command.
- If you wish to remove a row or a column, highlight it, then go to the right-click options and click on ‘Delete’.

4. Merging and Splitting Cells:

- In order to merge cells, one needs to first highlight the cells that one wants to merge, right-click, and click on the “merge cells” option.
- To split a merged cell, first select it, then go to the right-click option and choose the option that has “Split Cells.

- **Graphics:**

1. Inserting Pictures:

- Click on the ‘Insert’ tab in the ribbon and select ‘Pictures’ to insert an image from your computer database.
- Or, click on the ‘Online Pictures’ button to find and place images on the internet, such as Bing Image Search or OneDrive.

2. Formatting Pictures:

- Once you have placed the picture, you can format it using the picture tools tabs on the format and picture format toolbar.
- Modify the size/position of an object and wrap text. Write and draw styles, borders, and effects.

3. Inserting Shapes:

- Move to the “Insert” section and click on “Shapes” to insert shapes such as rectangles, circles, arrows, etc.
- By pointing at the document, the cursor changes to a cross, and drawing the shape requires that one click and drag it on the document. To format and customize shapes, you can use the Drawing Tools tabs.

4. Adding SmartArt:

- SmartArt is an option available in Word that can be used to make nice-looking graphics and designs.
- Click on the ‘Insert’ tab and then click on the ‘SmartArt’ from the available list of SmartArt graphic choices. To create the content of the SmartArt graphic, type text.

- **Wrapping Text Around Graphics:**

1. Positioning Options:

- Choose the image or shape, and in the Format or Picture Format tab, click on the “Wrap Text” option.
- Options include “In Line with Text,” “Square,” “Tight,” and so on to determine how the text will flow around the graphic.

2. Adjusting Text Wrapping:

- To adjust the text wrapping precisely, first place the cursor on the image or shape, right-click it and choose “Format Picture” or “Format Shape” from the drop-down list.

- If positioned in the Layout or Text Wrapping tab, one can set parameters such as the distance from the text, the tightness of the objects, or the alignment.

- **Knowledge Check 1**

  Fill in the blanks

  1. ___________ is a word processing software created by the Microsoft Corporation. (Microsoft Word)
  2. The word processing software that is popular with most people is Microsoft Word, which comes with a spell check and _____________ check. (grammar)
  3. The ________ tab is related to the general appearance of the paper and its organisation. (Page Layout)
  4. Wrap Text is used for full __________ text. (justifying)

- **Outcome-Based Activity 1**

  Open Microsoft Word, create a new document and then insert Pictures of your college.

7.4 Formatting Documents

Formatting a document in Microsoft Word enables the user to improve the appearance or structure of the information. Here's a comprehensive guide on formatting documents:

- **Basic Formatting:**

1. Font and Text Size:

- Highlight the text you would like to format, then select your desired font and change the size from the font drop-down list in the Home tab or the formatting bar.
- You can highlight the text using the options like bold, italic and underline.

2. Paragraph Alignment:

- Left-align or justify text and numbers with left and right alignment, respectively, centring it by going to the alignment tab or using the toolbar.

3. Line Spacing:

- Change the distance between multiple lines of text or between the paragraph and the next line by choosing options like single, 1. 5 lines or double spacing from the line spacing menu on the upper toolbar of the Word document.

4. Bullets and Numbering:

- A list can be created by using bullets and numbering that are available in the ribbon tabs. Adjust the list type and its appearance according to one’s preference.

- **Page Layout:**

1. Margins:

- You can change the page margins by clicking the “Layout” or “Page Layout” link in the ribbon and choosing the margins. You can select the standard scales or possible values or establish your own scales.

2. Orientation and Size:

- This is usually under the Layout tab under the File menu, where you can also change the page orientation (portrait or landscape) and size from the Page Setup options.

3. Headers and Footers:

- Add headers and footers to your document to indicate the number of pages, the title of the document, or any other data. The header and footer tools can modify these elements.

4. Page Breaks:

- Use page breaks to specify where a new page should start. You can do the page break from the ribbon by clicking the Page Break button, or you can manually insert the page break using the shortcut key Ctrl + Enter.

- **Advanced Formatting:**

1. Styles:

- Always use styles; if you don’t have standardised ones, you create your own to ensure that your formatting has a standard look. Some of the styles available are headed styles, titles, and the normal body style.

2. Tables:

- Use tables to group data and information in the writing process. Subheaders can be added to the table, and using the ribbon, one can also change the border, shading, and positioning of the table.

3. Images and Graphics:

- You can also add pictures and shapes to your document to make it more appealing. Shape, move, and change the image using the picture tools.

4. Columns:

- Subdivide your document into several columns to give it the look of a newspaper or a newsletter. Modify the settings for the respective columns from the Page Layout tab.

7.5 Software Add-Ons (Spell-Check, Hyphenation, Mail-Merge)

- **Spell-Check:**

Spell-Check is one of the Microsoft Word utilities designed to assist users in detecting miscalculations in the spelling of custom words. Here's how to use it: Here's how to use it:

1. Automatic Spell-Check: An added advantage of Word is that it has a spell check, which is enabled by default and checks for any spelling mistakes as you are typing. Words that are spelt wrong are generally marked with a red wavy line below it.
2. Manual Spell-Check: In the Word interface, to start checking for spelling errors, go to the tab labelled "Review." Then, in the ribbon bar, click on the "Spelling & Grammar" button or use the F7 shortcut on your keyboard. Word will also check the document's spelling and grammar and provide options for corrections.
3. Customising Spell-Check: Depending on the settings, users can add new words to the spelling list, switch between languages, and modify some proofing features.

- **Hyphenation:**

Hyphenation is a tool of Word that inserts hyphens in the words at the line ending to enhance the flow of the text and their readability. Here's how to use it: Here's how to use it:

1. **Automatic Hyphenation:** Word can also hyphenate text for you either as you are typing or when you align the text in the document. This avoids the formation of ugly line breaks and makes the text look better.
2. **Manual Hyphenation:** Users can manually hyphenate individual words or a part of the text by selecting the text, going to the tab "Page Layout", clicking on "Hyphenation," and selecting "Manual. " Word will then suggest where hyphenation can occur, and then the user can set the hyphenation to the desired place.

3. **Customising Hyphenation:** The user has some control over how the word wrapping and line breaking is done through the hyphenation zone width, where hyphenation is allowed, or even if the program is to hyphenate the text.

- **Mail Merge:**

Mail Merge is a Microsoft Word tool that enables users to create letters, envelopes, labels, or even emails, all customised using data from a database or spreadsheet combined with a template. Here's how to use it: Here's how to use it:

1. Prepare Data Source: First, you can begin with a new or existing document, such as an Excel sheet or Outlook contacts list, containing the data to be merged into the document (e.g., names, addresses, and email addresses).
2. Create Document Template: Create the document layout in Word, in other words, write down the text of the document and create special fields (merge fields) where the data from the data source should be inserted. These placeholders are usually enclosed in chevrons << >>.
3. Start Mail Merge: Click on the “Mailings” tab in the ribbon and click on “Start Mail Merge. ” Choose the kind of document you wish to create (letters, envelopes, labels, etc.) and select the document layout.

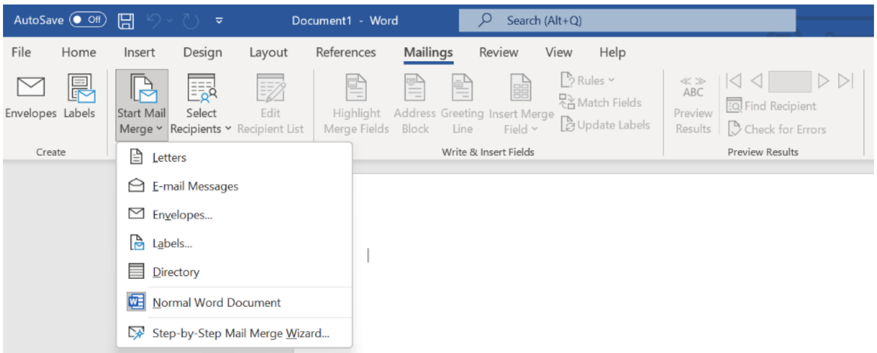


Fig: Mail Merge

4. Insert Merge Fields: Within the document template, insert merge fields that state the point at which the content of the data source is to be incorporated.
5. Preview and Complete Merge: This will allow you to view all the merged documents and ascertain whether everything looks good. Then, perform the merge to get separate documents for every record in the data source.
6. Print or Send: After the merge is done, users can only print the documents or send them directly as e-mail if the merged document is an e-mail document.

7.6 Printing Documents, Envelopes, and Labels

It is quite easy to print documents, envelopes, and labels in the Microsoft Word program as this tool is integrated into the program. Here's a guide on how to print each of these items:

Printing Documents:

1. Open the Document: In the Microsoft Word application, select the document you want to print from the list.
2. Review the Document: After completing the content of the document, it is good to go through the document and check if everything is well positioned and if there are mistakes.
3. Access Print Options: Click on the “File” tab located at the upper left corner of the window, then click on “Print. ” You can also use the keyboard shortcut Ctrl + P.
4. Set Printer and Settings: In the Print dialogue box, you have a list of available printers. Select your printer from among them. You can also change other print options, including the number of copies, the pages to be printed, the type of paper to be used, etc.
5. Print: To print the document, click the “Print” button. Wait for the printed characters to come out.

Printing Envelopes:

1. Open the Envelope Template: In Microsoft Word, click at the top, look for the “Mailings” tab and click on “Envelopes. ”
2. Enter Recipient Information: In the Envelopes and Labels dialogue box, for a domestic address, you need to type the recipient's address in the option “Delivery address.” You can also include your return address, depending on any need that may arise as you send your documents.
3. Select Envelope Size: Select the size of the envelope from the drop-down list labelled “Envelope size”. You can also change the size of the envelope by going to “Options. ”
4. Preview and Adjust: To view how the envelope will appear, click the “Preview” button. Then, you must make any necessary corrections to the format and position of the text.

5. Print: After filling out the form, click the ‘Print’ button to send the envelope to the printer. Make sure that the envelope fits properly on the printer tray, following the printer’s recommended instructions.

Printing Labels:

1. Open the Label Template: If you are using the Microsoft Word format, then on the “Ribbon,” click on the “Mailings” tab, and then click on “Labels.”
2. Enter Label Information: To do this, in the Labels dialogue box, type the text that should be printed on the labels in the Address field. You can also get data from an Excel or any other data source of your choice and then import it into the program.
3. Select Label Type: Select the label type you are using from the label vendors list and the product number drop-down list. If your label type is not listed, please go to “New Label” to define your desired label size.
4. Preview and Adjust: To check how the labels will look, click “Preview.” Do not forget to adjust the paragraphs if their format or location does not fit the rest of the paper.
5. Print: Once done, go to the top-right of the webpage and click the “Print” button to print the labels. Ensure that the label sheets are loaded in the printer tray appropriately, as recommended by the printer manufacturer.

- **Knowledge Check 2**

  Fill in the Blanks

  1. __________ are the documents or files that are already prepared and from which new documents in MS Word can be generated. (Templates)
  2. ____________ in MS Word is a macro that records a set of operations that can be repeated when a button is pressed. (Macro)
  3. _________can be used in the organisation of large-scale sending of similar messages or the preparation of messages containing individual details of targeted recipients. (Mail merge)

- **Outcome-Based Activity 2**

  Students should have a look at the possible use of mail merge in the context of business letters, invitations, etc.

7.7 Document Collaboration Features

Microsoft Word has several features that allow collaboration, such as more than one user working on the same document at the same time. Here are some of the key collaboration features available in MS Word: Here are some of the key collaboration features available in MS Word:

1. Real-time Co-Authoring: Word is a collaborative application that allows users to edit the same document simultaneously. Work done by one user can be seen by others simultaneously, implying the continuity of users' cooperation.
2. Comments and Track Changes: Comments and changes can be included in the document so that users can easily review them. Instead, comments can be replied to and resolved, and changes can be accepted or declined one by one or in a batch.
3. Version History: There is an option for version history. The Word program will save different versions, and the user can always go back to a previous version in case they wish to continue from where they left off. This feature allows some changes to be made and, if they were not well received, turned down.
4. Permission Settings: Users can define the access level of other people to view, edit, or comment on the document. Permissions may be personal or for a certain group of collaborators.
5. Integration with OneDrive and SharePoint: Word also synchronises with OneDrive and SharePoint from Microsoft, which means that the user can save documents online and work with others from any location on any device.
6. @ Mentions: Both in the comments section and within the document, by typing '@' followed by the name or email of the collaborator, users can tag them. This feature informs the mentioned user and also assists in drawing the user's attention to a particular section in the document.

7.8 Summary

- Microsoft Word is a commonly used word processor that can be employed to write and edit text and format documents. It has a simple design and is perfect for anything, from creating an article to writing an essay or making a report or any professional document.

- The basic operations which one can perform in MS Word include starting a new document, opening a document, saving the document for the first time, saving a document, entering text, formatting margins, headers and footers, putting numbers on the pages and exiting from a document. These tasks are critical in document production, modification, and archiving.
- Mail merge is useful when one wants to produce a large number of letters, emails or labels from a main document and a data source. Macros, on the other hand, are used to automate the most used sequence of commands or actions in order to save time.
- Templates allow users to have different types of documents, such as resumes, newsletters, invoices, etc., so users do not need to prepare documents on their own.
- It is an effective word-processing application that has many features and functions to help the user compose, modify, and design text documents. Mail merge and macros are some of the facilities that are incorporated, which help increase productivity, and templates that assist in creating pre-design documents.

7.9 Keywords

- **Microsoft Office application:** Microsoft Office includes several applications commonly used in the business environment, including Word, Excel, and PowerPoint.
- **User Interface (UI):** This is the interface between the user and the software or application being used; hence, users are able to manipulate it.

7.10 Self-Assessment Questions

1. What is the significance of Microsoft Word as a word processing software?
2. How can you start Microsoft Word and begin working on a new document?
3. What are the different Ribbon Menu categories in MS Word, and what functionalities do they offer?
4. What are the primary tasks in MS Word, and how do they contribute to document creation and management?
5. How does mail merge to enhance the functionality of MS Word, and what are their specific uses in document processing?

7.11 References / Reference Reading

- Lambert, Joan. "Microsoft Word Step by Step (Office 2021 and Microsoft 365)." Microsoft Press, 2021.
- Cox, Joyce, and Joan Lambert. "Microsoft Word 2019 Step by Step." Microsoft Press, 2018.
- D'Souza, Janelle, and Amit Kalra. "Mastering MS Word: Beginner to Advanced." BPB Publications, 2022.
- Saxena, Sanjay. "MS Office: A Practical Approach." Vikas Publishing House, 2020.
- Goel, Anita. "Computer Fundamentals and Office Automation." Pearson Education India, 2019.

Unit 8: Advanced Features of MS Word

Learning Outcomes:

- Students will be able to define styles and templates in MS Word.
- Students will be able to explain the process of using styles and templates.
- Students will be able to demonstrate how to create and apply macros.
- Students will be able to utilize collaboration features like track changes and comments.
- Students will be able to create documents with advanced formatting and table features.

Structure:

8.1 Styles and Templates

Styles in Microsoft Word are predefined formatting settings that allow you to apply consistent formatting throughout your document. Here's a comprehensive guide on how to work with styles:

Applying Styles:

1. Applying Built-in Styles:

- Select the text you want to format.
- Go to the "Home" tab in the ribbon.
- In the "Styles" group, you'll see a gallery of built-in styles such as Heading 1, Heading 2, Normal, etc. Click on the style you want to apply.

2. Applying Styles from the Styles Pane:

- Click on the dialogue box launcher in the Styles group (the small arrow in the bottom right corner).
- The Styles pane will open and display a list of styles. Click on the style you want to apply to the selected text.

3. Clearing Styles:

- To remove formatting and revert to the default style (usually "Normal"), select the text and choose "Clear Formatting" from the Styles group.

Creating and Modifying Styles:

1. Creating a New Style:

- Select the text with the formatting you want to save as a new style.
- Right-click on the selected text and choose "Styles" > "Save Selection as a New Quick Style."
- Enter a name for the new style and click "OK" to save it.

2. Modifying Styles:

- Right-click on the style you want to modify in the Styles pane and choose "Modify."
- Make changes to the formatting options (such as font, size, colour, etc.) in the Modify Style dialogue box.
- Click "OK" to save the changes.

3. Deleting Styles:

- To remove a custom style, right-click on the style in the Styles pane and choose "Delete."
- Be cautious when deleting styles, as this action cannot be undone.

Using Styles for Document Formatting:

1. Heading Styles:

Headings and subheadings, such as Heading 1, Heading 2, etc., should be used to organise your document. These styles create a hierarchical and mandatory organisational framework for producing a table of contents.

2. Body Text Style:

Set the body text to the "Normal" style. This helps keep the font type, size, and spacing of the document uniform.

3. Custom Styles for Emphasis:

Design different classes of text that need to be styled with some degree of distinctiveness from other text, or that is somehow different from the main body of the text.

Templates in MS-Word

Templates are convenient plans created beforehand with MS Word and used as the basis for creating documents. Such tools offer a format, arrangement, and design for the documents, which can be modified in accordance with organisational needs to reduce the workload of generating the documents.

MS Word also provides a large number of templates for different fields and types of documents, such as resumes, cover letters, business letters, brochures, flyers, invitations, and many more. These templates are pre-designed to professional standards and can be easily located through the "New Document" window or in the MS Word "Templates" folder.

With templates, users can create rather similar documents in terms of design and formatting. It makes it easy for users to feed in their content while having a professional outlook. Templates may contain blanks for text, images, tables, or any other content, it is simpler to manipulate the document.

Besides the standard templates that come with the software, one can still make their own templates. This enables them to set the structure and appearance of the files that they want to use in the future and which can be repeated. Custom templates are most helpful in organisations that need to maintain a certain degree of branding or use certain document formats.

For instance, in MS Word, to apply a template, a user picks a template of their choice, and then the program will develop a document for them from the said template. Users are also free to edit the content and layout of the content and remove or include some of the items and features. The first type of template is the master copy that is used for replication for subsequent use in the future.

The author notes that using templates in MS Word is a convenient means of developing good documents quickly. They save time, as one does not have to begin from the barest of beginnings and form the basis of numerous types of documents. For individual or business use or even for certain projects, templates in MS Word enable easy document creation and, at the same time, present documents that look professional.

8.2 Macros and Automation

Macros:

Macro in MS Word is one of the automation tools that can record a sequence of operations and then execute them in a single step. They are applied to reduce the time and effort involved in a particular task and make processes more effective and efficient. In simple terms, a macro is defined as a sequence of instructions or a command that a single action can run. Then, users can record their actions, such as formatting text, applying a style, inserting tables, and performing other intricate operations; then, the actions can be saved in a macro. The macro can then be saved and linked to a button, or a keyboard shortcut for convenience.

A macro, on the other hand, enables the user to automate a process that would otherwise need a person to intervene. For instance, they can develop a macro to align a document to some standard, format it to apply style throughout the document, or do calculations on a large set of data. Macros can come in handy for most operations that are repeated frequently, or take a relatively long time to complete, as they will help significantly reduce the amount of time spent on such operations.

MS Word also offers a good user interface for creating, managing, and editing macros to automate work. Users can record macros on the fly, edit recorded macros, create new macros using VBA, and save them with a button or shortcut key.

It should be noted that macros can be really effective, but one should be careful when using them. Macro can be used to carry out unwanted actions; therefore, macro security settings should be enabled, and the macros should be run from secure sources.

Automation

Automation in Microsoft Word refers to the tasks that are accomplished utilising different features and tools of the program chiefly for the purpose of saving time. Here are key aspects and methods for automating tasks in Word: Here are key aspects and methods for automating tasks in Word:

1. Macros

What are Macros?

- Macros are exactly predefined sets of actions that should be performed in Word several or tens of times.
- This is a script written by using visual basic for applications or (VBA).

Creating a Macro:

1. Go to the "View" tab, click on "Macros," and select "Record Macro."
2. Name your macro and assign it to a button or keyboard shortcut.
3. Perform the tasks you want to automate while Word records your actions.
4. Stop recording by clicking "View" > "Macros" > "Stop Recording."
5. Run your macro by going to "View" > "Macros" > "View Macros," selecting your macro, and clicking "Run."

Editing a Macro:

1. Go to "View" > "Macros" > "View Macros."
2. Select your macro and click "Edit" to open the VBA editor.
3. Modify the code as needed to refine or expand the macro's functionality.

2. Quick Parts

What are Quick Parts?

Quick Parts are reusable pieces of content, such as text snippets, images, and tables, that you can insert into documents quickly.

Creating Quick Parts:

1. Select the text or object you want to save as a Quick Part.
2. Go to the "Insert" tab and click on "Quick Parts" in the "Text" group.
3. Choose "Save Selection to Quick Part Gallery."
4. Name your Quick Part, select a gallery (e.g., AutoText), and click "OK."

Using Quick Parts:

1. Place your cursor where you want to insert the Quick Part.
2. Go to "Insert" > "Quick Parts" and select the desired Quick Part from the gallery.
3. Templates

What are Templates?

Templates are pre-designed documents that serve as a starting point for new documents. They include predefined styles, formatting, and content.

Creating a Template:

1. Design your document with the desired styles, formatting, and placeholders.
2. Go to "File" > "Save As."
3. Choose "Word Template" from the "Save as type" dropdown menu and save your template.

Using a Template:

1. Go to "File" > "New."
2. Select "Personal" to find your custom templates or browse for available templates online.
3. Choose the desired template to create a new document based on it.

4. Mail Merge

What is Mail Merge?

Mail Merge is a feature that allows you to create personalised documents (e.g., letters, labels, emails) by merging a Word document with a data source (e.g., Excel spreadsheet).

Using Mail Merge:

1. Go to the "Mailings" tab and click on "Start Mail Merge."
2. Choose the type of document you want to create (e.g., Letters, Labels).
3. Select "Recipients" to choose your data source (e.g., an Excel file).
4. Insert merge fields into your document where you want personalised information to appear.
5. Preview the results and click "Finish & Merge" to generate the merged documents.

5. Automatic Text Formatting

AutoCorrect and AutoFormat:

- AutoCorrect automatically corrects common typing errors and replaces predefined text with specified entries.
- AutoFormat applies automatic formatting to text based on predefined rules.

Using AutoCorrect:

1. Go to "File" > "Options."
2. Select "Proofing" and click on "AutoCorrect Options."
3. Customize the AutoCorrect entries as needed.

Using AutoFormat:

4. Go to "File" > "Options."

5. Go to the “Proofing” tab, and click on “AutoFormat As You Type. “
6. In AutoFormat settings, you should change the options to the desired one depending on your choice.

- **Knowledge Check 1**

Fill in the Blanks

1. __________ are documents or files that are already created, and act as templates that the user uses to create a new document in MS Word. (Templates)
2. ____________ in MS Word are a strong macro instrument that enables a person to record a sequence of activities and play it as one command. (Macros)
3. __________ is used in organisations to reduce the number of mailings that need to be sent out in large quantities or to address individual letters containing unique information for each recipient. (Mail merge)

- **Outcome-Based Activity 1**

Students will talk about how mail merge can be used in different scenarios, preferably in business or in event invitations.

8.3 Collaboration Features (Track Changes, Comments)

Tools present in Microsoft Word and aid in collaboration include Track Changes and Comments, which allow multiple users to review and work on a document simultaneously. Here's a comprehensive guide on how to use these features: Here's a comprehensive guide on how to use these features:

- **Track Changes:**

1. Enabling Track Changes:

- Locate the document for collaboration in Microsoft Word that you would like to edit and open it.
- To do this, navigate your cursor over the “Review” tab within the ribbon that is positioned at the top of the screen.
- To turn on the Track Changes feature, click Track Changes on the Review tab’s Compare & Review group. Usually, when Track Changes is on, all changes performed in the document will be marked and highlighted.

2. Reviewing Changes:

- Any changes made to the document will be highlighted in different colours to distinguish the last modifier of the document.
- Single change review is done by using the “Prev” and “Next” buttons in the Tracking group, to move up and down through the document.

3. Accepting or Rejecting Changes:

- To accept a proposed change, position the cursor at the edited text and select Accept from the Tracking group.
- To reject a proposed change, position the pointer in the edited text and click Reject in the Tracking group.

- **Comments:**

1. Inserting Comments:

- To insert a comment, highlight the text to which the comment will be applied or move the cursor to the application point.
- On the Review tab, in the Comments group, select New Comment.
- You will type your comment in the comment pane, a box opened at the right side of the document.

2. Replying to Comments:

- If another person has already posted a comment, click the ‘Reply’ button at the right of the comment section in the comment area to reply to the comment.

3. Resolving Comments:

- When a comment has been dealt with or replied to, the user can then go ahead and Flag the comment as complete by going to the ‘Comment’ tab, right-clicking on it and clicking ‘Resolve Comment’.
- Deleted comments will be collapsed by default, though scroll down to the Comments section to click on the link, “Show Comments.”

4. Navigating Comments:

- For the movement with the comments within the document, you can use the “Previous Comment” and “Next Comment” buttons in the Comments group.

- **Protecting Document Changes:**

1. Locking Changes:

- If you don’t want other users to add modifications to the document, you should click on the “Lock Tracking” in the Tracking group.

- You can also limit the editing by accessing the “Review” tab and clicking the “Restrict Editing” button, and then setting the necessary editing restrictions and entering a password if any is to be set.

8.4 Advanced Table Features

In Microsoft Word, other features that concern the tables enable the user to modify and format tables depending on personal preference. Here's a comprehensive guide on some of the advanced table features available: Here's a comprehensive guide on some of the advanced table features available:

- **Table Design:**

1. Table Styles:

- Word has different predefined table styles that you can use to format tables; changing their style is very easy.
- To access different table styles and make further changes, go to the “Design” tab in the ribbon.

2. Borders and Shading:

- Format the table's borders and shading in the “Borders” and “Shading” pull-down menus in tabular form.
- Borders can be applied to a cell, row, column or the entire table if you want.

3. Layout Options:

- Regarding cell positioning, there is a feature known as Layout Options, which can be found in the “Layout” tab and allows for control over the disposition of cells in the table.
- Modify the cell margins, cell spacing, and text wrapping to achieve the desired table format.

- **Advanced Table Manipulation:**

1. Splitting and Merging Cells:

- How to use the Split Cells tool is to divide the cells into two or more cells, and the Merge Cells tool is to join the cells into one cell.
- To split or merge cells, you must first highlight the cells you wish to modify, right-click on the selection, and choose the required operation from the menu.

2. Sorting and Filtering:

- To sort the table data, right-click on the table header and select ‘Sort Ascending’ or ‘Sort Descending’ depending on the requirement or use the drop-down button on the right-hand side of the chosen table header and click on Sort.
- It is also possible to apply filters to table columns to quickly filter particular subsets of data.

3. Formulas:

- You can even include calculations like in Excel right inside the cells of the table.
- Place a formula using a cell, then navigate to the “Layout” tab and select “Formula”. Type the formula in the cell using the cell reference signs along with the mathematical signs.

- **Table Properties:**

1. Row and Column Properties:

- You can increase or decrease the height of a table row on the row level and change the width of a table column on the column level by clicking on the respective line and dragging it up or down, respectively.
- There is also an option to select a row or a column click and then select table properties to set a specific size.

2. Cell Properties:

- To align text or text in individual cells with each other, click on the cell, go to cell properties, and then select the required cell property like text direction, text vertical alignment, and text direction.

- **Table Tools:**

1. Table Tools Tabs:

- When you select a table, two additional tabs appear in the ribbon: “Design” and “Layout. ” These two tabs have extra features and functions that are useful when handling tables.

2. Draw Table Tool:

- Drawing Table—This tool allows you to draw a table on a page and create a table of the desired type by choosing the necessary cells. It is useful for creating tables with complicated shapes or non-rectangular angles.

- **Knowledge Check 2**

  Fill in the Blanks

1. To merge a cell in a Word table into a single cell, you use the _____ option found under the Table Tools Layout tab. (Merge Cells)
2. If you have a wide table that spans multiple pages, you might want the first or the header row to be at the top of every page, so to do that, you should ________. (repeat header rows)
3. If you want to combine several cells into one cell in a Word table, you select the cells that you want to combine and go to the ________ option in the Table Tools Layout tab. (Merge Cells)
4. To arrange the data in a table, there is an _________ button at the Table Tools Layout tab, which allows you to sort the data either alphabetically or numerically. (Sort)

- **Outcome-Based Activity 2**

  Start Microsoft Word, insert a simple table with sample data, and use the following:

  - Splitting Cells: Illustrate how you can convert a single cell into several cells.
  - Merging Cells: In this way, shows how multiple cells can be merged into one.

8.5 Advanced Document Formatting

Formatting of documents in the latest MS Word enables one to prepare high-quality papers with special layouts, styles, and designs. Here are some advanced formatting features in Word:

1. Styles: Word contains formatted styles for different items like heading, title, sub-title and paragraph. There are also style groups, and you can create your style to maintain a certain format throughout the document. They assist in the repetition of formatting changes and guarantee that parts of the document blend in with each other.
2. Themes: Word styles are a well-coordinated collection of colours, fonts, and effects that can be used throughout the document. Themes are useful in generating more attractive documents that are easily identifiable due to the uniformity of the designs employed on them.
3. Columns and Sections: Word enables one to arrange the document in different columns or sections with various designs. This is helpful for any newsletter, brochure, or document that has other complicated layouts to fulfil.

4. Page Layout Options: Word provides features for positioning the page's contents, such as margins, orientation, size, and breaks. Headers and footers, such as page numbers, dates, and document titles, can also be customised.
5. Table of Contents: From Word, a user can easily display the table of contents depending on the headings and styles used in the document. A variety of styles can be applied to the appearance of the table of contents, and one can revise it instantaneously when changes are made to the document.
6. Advanced Text Formatting: Word provides formatting tools in a text that enables one to control the characters and space between them, the use of special characters and the effects to be applied to the text. The other features that users can use include drop caps, small caps and other features that enhance the typographic features and better read and attractive appearance.
7. Graphics and Multimedia: Word also enables users to insert and format images, shapes, charts, and SmartArt graphics in the document. It allows users to include multimedia content like videos and audio files.
8. Watermarks and Backgrounds: Word helps set watermarks and background images or colours for the document in a bid to make it more attractive and consistent with the company's brand.

8.6 Summary

- Templates in formatted documents such as MS Word are layouts that enable the user to use the same format throughout the document.
- Templates are pre-established documents or files that act as a basis for developing new ones in MS Word. They offer conventions regarding how information should be arranged and presented and which writing styles can be used. They are always ready to be adapted to certain uses, making work much more efficient in terms of document construction.
- Macros in MS Word is a tool that records a series of operations and then shifts the recorded operation as a single command.
- Automation in Microsoft Word involves applying numerous facilities and deliveries to accomplish similar operations, simplify business processes, and improve efficiency.

- Mail Merge is an option that enables users to produce documents with individual contents (for instance, letters, envelopes, emails) by incorporating data from another source such as Microsoft Excel.
- For major operations, macros are used, while for specific contents, Quick Parts are used. Standard papers are templates; personalised letters are mail merges, and text corrections are autocorrect and auto format.

8.7 Keywords

- **Templates**—templates are documents or files already created and used as a guide when creating another document in MS Word.
- **Mail Merge—Mail Merge is an option that allows you to generate custom documents (for example, letters, labels, and** email messages) by integrating the document created in Microsoft Word with another file, usually a database in Microsoft Excel.
- **Macros**—A macro in MS Word is a special tool that records the sequence of actions and allows you to replay them with a single command.

8.8 Self-Assessment Questions

1. What are Templates?
2. What are Macros?
3. What is Mail Merge?
4. What are Advanced Table Features?
5. Describe Collaboration Features in MS Word.

8.9 References / Reference Reading

- Kogent Learning Solutions Inc. *MS Word 2016 in Depth.* Dreamtech Press, 2017.
- Jain, Satish, and Shashank Jain. *Comdex Professional Edition Microsoft Office 2016 and 2019: A Comprehensive Approach.* Dreamtech Press, 2019.
- Lal, Ramesh Bangia. *Learning MS Office 2016: A Step-by-Step Approach.* Khanna Book Publishing Co. Pvt. Ltd., 2018.
- Vermaat, Misty E. *Microsoft Word 2019: Comprehensive.* Cengage Learning, 2019.

Unit 9: Introduction to MS Excel

Learning Outcomes:

- Students will be able to identify components of workbooks and worksheets.
- Students will be able to explain how to open and save workbooks.
- Students will be able to demonstrate the use of basic formulas and functions.
- Students will be able to apply cell and data formatting techniques.
- Students will be able to utilise sorting and filtering tools for data analysis.

Structure:

9.1 Understanding Workbooks and Worksheets

Excel is a tool used especially for analysing data within a spreadsheet where data is entered, managed, and processed. Another feature is the Ribbon, a graphical toolbar that is used to access most of the commands in the Excel application. The Ribbon is a fundamental tool in MS Excel that allows for the various choices and features to be found, and this knowledge is mandatory.

The Ribbon in MS Excel has subgroups, each related to a particular function or operation that needs to be performed. The main tabs are Home, Insert, Page Layout, Data, and many others; every tab is presented as a set of command buttons related in terms of their functionality, which helps a user find necessary options and features faster.

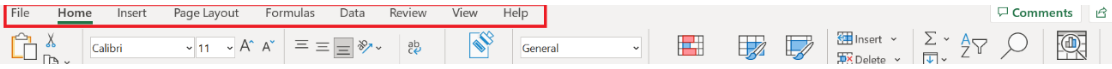


Fig: Ribbon in MS Excel

The Home tab is in the Ribbon section and contains tools for data formatting. It enables users to change the font style, align the contents of the cell, format numbers, and perform simple operations such as sorting and filtering the data. It also offers the possibility of quickly accessing frequently used commands like copy, paste, and undo.

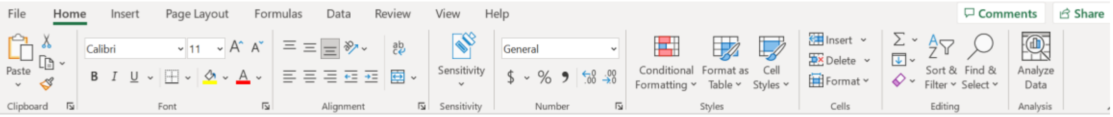


Fig: - Home tab

The Insert tab includes additional options for incorporating various objects into a spreadsheet. You can insert tables for data sorting, charts for displaying tendencies, pictures and shapes for the document's decoration, and other objects like hyperlinks, headlines, and footers for the document's construction.

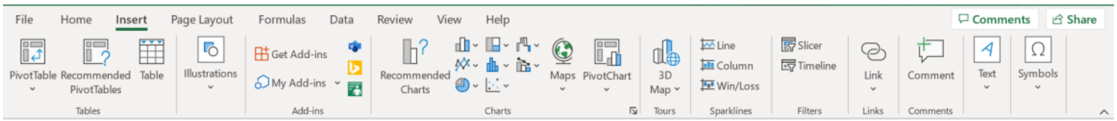


Fig: - Insert tab

The Page Layout tab concerns the appearance of the spared sheet. It allows for setting up margins for the page, choosing the orientation of the page, defining the header and footer of the page, and applying themes to the document to make it look professional. Users are also able to choose the type of printout and even preview the format before opting to print.

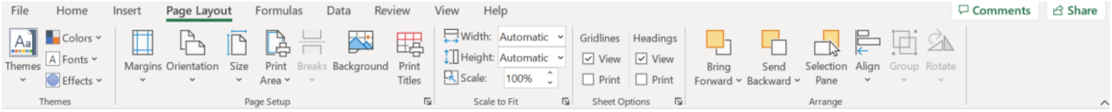


Fig:- Page Layout tab

Microsoft Excel's Data tab is a core part of the program used to manipulate data. It provides the possibilities for importing and exporting data from other sources, excluding copies, checking the data inputs, and performing some data analysis operations, such as creating data tables and consolidating data.

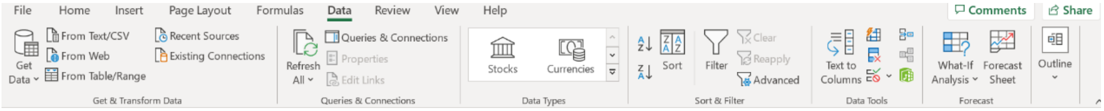


Fig: - Data tab

The Review tab is used to review the changes made in the spreadsheet and edit options. It has features such as spell-checking, cell commenting, protecting the workbook or individual sheets with passwords, and tracking changes made by different users. Another useful tab is More, which is convenient when working in a team and when it is important not to lose data.

9.2 Opening and Saving Workbooks

Opening and Saving Workbooks in Microsoft Excel

To open workbooks in Microsoft Excel, you can easily do so by double-clicking the Excel file or by right-clicking and selecting open, as well as clicking on the Excel file and choosing the open option to open workbooks; you can use the save option to save the workbook as well.

Opening a Workbook

1. Using the File Menu:

- Open Excel: Open Microsoft Excel by double-clicking the Excel icon on your computer or by clicking Start and then searching for Excel.
- Accessing the File Menu: After you have opened Excel, go to the top left-hand side of the screen and click on the 'File' tab.
- Open Option: In the drop-down list that appears, choose the 'Open' option. This will show a list of the files that were recently opened and other options for searching for the file.
- Browse for File: If the workbook that you wish to open is not in the list, click on 'Browse', and this will open the directory of File Explorer through which you can

locate the particular workbook. Click to choose the appropriate workbook and then click on ‘Open.

2. Using the Keyboard Shortcut:

- Shortcut: The ‘Open’ dialogue is also opened by using the keyboard shortcut control and O. This will immediately open a folder that leads you to File Explorer, where you can select the particular workbook you wish to open.

3. Opening from Recent Files:

- Recent Workbooks: In Excel, there is an option that allows you to use the workbooks easily opened recently. The ‘Recent’ part of the ‘File’ tab gives you a list of the workbooks that you have most recently used. To view any file in this list, click on that file.

4. Drag and Drop Method:

- Drag and Drop: The other easy approach is to open the Excel application, hold the Excel file, right-click, select copy, and then paste the file either by hitting the control and V keys simultaneously or by moving the mouse to the desired location on the open Excel application window, holding the left mouse button, and then releasing it. The workbook should be opened automatically.

Saving a Workbook

1. Using the File Menu:

- Save As To save a new workbook or a workbook with a different location and name, go to ‘File’ and then ‘Save As’. Specify the folder location, enter the workbook name, and then click on ‘Save.’
- Save: If you want to save changes to an already existing workbook, then you go to the ‘File’ tab and click on ‘Save’, or you can press the shortcut key on the keyboard, ‘Ctrl + S’.

2. Saving to OneDrive:

- Cloud Storage: Microsoft Excel allows you to save workbooks directly on OneDrive, Microsoft’s cloud storage system. In the ‘Save As’ mode, choose OneDrive and enter your Microsoft credentials. Save the workbook in the desired folder, then click the ‘Save’ button.

3. AutoSave Feature:

- Automatic Saving: Excel has the AutoSave feature for the workbook saved on OneDrive or SharePoint by default. AutoSave is active when this option is on, and Excel saves your modifications in a few seconds. This feature assists in avoiding loss of data, and it auto-saves your work to prevent a lot of data input in case of power failure or any other factor that may cause interruption of the data input process.

4. Exporting as Different Formats:

- Different Formats: At other times, you may be required to export the workbook in a format that is different from the default format and instead use PDF or CSV format. To do this, click on 'File,' then 'Save As,' and from the 'Save as type' drop-down list, select the format you want before clicking on 'Save.'

5. Saving a Copy:

- Duplicate Workbook: If you require a copy of the workbook, navigate to 'File' -> 'Save a Copy.' This enables the user to save the current version of the workbook with a different name or in a different directory, and this will not affect the original copy of the workbook.

- **Knowledge Check 1**

  Fill in the Blanks

  1. Excel provides _________ features to enable the software to be used by visually or physically challenged persons. (accessibility)
  2. The approach of Excel is based on the broad set of _________ tools that are implemented within the application to assist the users in the process of data analysis and interpretation. (data analysis)
  3. The __________ option that is included in Excel helps correct spelling mistakes or typos every time a person is typing. (AutoCorrect)

- **Outcome-Based Activity 1**

  Explore Themes and Styles in MS EXCEL.

9.3 Basic Formulas and Functions

Functions are one of the most important features of Microsoft Excel software, and they help enable calculations, data manipulation and management, and cell interdependence.

As with any formula, there are settings in Excel that can be adjusted to enhance the function's performance. These settings affect formula behaviour, error detection and the degree of rounding off. Let's explore some of the key settings for formulas in MS Excel:

Formulas in MS Excel, along with a short description for each:

1. SUM Formula: The SUM formula is applied to obtain the sum of the selected cells. It sums up the values in the stated cells and gives the total. For example, "=SUM(A1:A10) calculates the total of the cells A1 through A10.
2. AVERAGE Formula: The AVERAGE formula helps to find the mean of the range of cells. It sums up the values in its arguments and divides the result by the count of cells in the range. For instance, "=AVERAGE(B1:B5)" divided the values in the cells B1 to B5 and gave the result of the average of the values in the cells.
3. IF Formula: The IF formula enables one to perform conditional calculations. This type of function tests a specific condition and gives one result if the condition is true and another result if the condition is false. For instance, "=IF(A1>10, "Yes", "No")" tests whether the value within cell A1 is greater than 10. If true, it gives out "Yes"; if otherwise, it gives out "No".
4. VLOOKUP Formula: The VLOOKUP formula is an Excel formula which searches for a particular value in the first column of the range and returns the value for that particular cell in the second column. It is frequently applied for data search and information retrieval. For instance, "=VLOOKUP(A1, C1:The formula "=ISNUMBER(SEARCH("A1", CHAR(67)+ "C1:D10, 2, FALSE)" searches for the value of A1 in the cells C1 to D10. If found, it goes to the range of cells from C2 to C10 and returns the value of the row found in the second column.

Functions in Excel – Average, MIN; MAX, COUNT IF, Function arguments

Built-in functions in Excel are templates that have been developed in advance to perform tasks specific to data analysis. They are intended to reduce the load of complicated computations and tedious work, of which data analysis and manipulation are no exceptions. More importantly, Excel provides a number of functions that one can use to meet a certain need, such as mathematical computations, statistical analysis, date and time functions, and text functions, among others.

Excel functions are created in a particular way in that the equal sign (=) is used, followed by the function name with arguments enclosed in brackets. Arguments can be values or references to other cells in the spreadsheet or other functions. The Excel functions can be divided into several types depending on the purpose of their use; these types can include mathematical functions (for example, SUM or AVERAGE), logical functions(for example, IF, AND or OR), textual functions(for example, CONCATENATE, LEFT OR RIGHT) among others.

Functions in Excel are among the most important elements of the spreadsheet. They enable the user to carry out many operations and manipulate the data. They are formulas that integrate into the program, make complex work easier, and avoid using a scalpel.

In Excel, there are so many functions that one can use to suit a given need or task. Some common functions include:

- SUM: Sums a set of values.
- AVERAGE: Defines the measure of the central tendency of a given set of data.
- MAX: This function will return the maximum of a range.
- MIN: This function gives the minimum value in a range.
- COUNT: Returns the count of the cells in the active range or sheet that contains values.
- IF: Performs a logical check and yields other results depending on the result maintained by a program.
- VLOOKUP: Searches a value in a table and then retrieves a value from another column. 1 (Program: MS Excel, Contributor: Curtis D. Fyer)
- CONCATENATE: Joins a number of text strings into a single string.
- DATE: Creates a date by concatenating the year, month and day variables.
- ROUND: Rounds a number given to the number of decimal places given as the input.

The above are some of the many functions in Excel that enable performers to accomplish their tasks. Each of them has its syntax and possible arguments defining its actions. Functions in the formula bar can be used directly or selected by the user from the Function Library in the ribbon.

Excel users also have the option to create unique functions using Visual Basic for Applications, a programming language within Excel. VBA allows the user to create his/her own function to perform certain calculations or even control the execution of certain activities.

Functions in Excel are of immense use in analysing data, modelling, statistical computations, and so on. It allows users to save time, increases precision in data handling, and increases the effectiveness of using data in spreadsheets. Applying functions, users are able to maximize the potential of Excel and perform a lot of operations effectively.

9.4 Formatting Cells and Data

Formatting Cells

1. Accessing Cell Formatting Options:

- Home Tab: The' Home' tab on the ribbon offers almost all the formatting options for cells: font type, size, colour, borders, and alignment.
- Format Cells Dialog Box: If more exquisite formatting options are required, choose 'Format cells' by right-clicking on the cell or using hotkeys—Ctrl + 1 to open the Format Cells dialog.

2. Font and Text Formatting:

- Font Style and Size: To change the font style and size, select the options from the drop-down boxes available in the Home tab. These are options: the fonts include the first one in bold, the second one in italics, and the last one in underlined.
- Font Colour: To change the text colour, click on the button labelled 'Font Colour' and then the 'A' with the coloured bar beneath it.
- Text Alignment: To master the alignment of text within a cell, you must use the alignment features. You can do this horizontally (left, centre, and right) and vertically (top, middle, as well as at the base).

3. Number Formatting:

- General Number Formats: Use the 'Number' group in the Home tab to select the formatting as currency, percentage, date, or time.
- Custom Number Formats: The Format Cells dialog box has a 'Number' tab that helps create number formats. This is helpful for particular data displays, including numbers in scientific notations or fractions.

4. Cell Borders and Fill:

- Borders: Additional or editing of cell borders can be done using the 'Borders' button found in the Home tab. Below the main selection, you can pick the border type and its colour as well.
- Fill Colour: To change the background of a cell, click the 'Fill Colour' button depicted by a paint bucket icon.

5. Conditional Formatting:

- Highlighting Rules: Conditional formatting is used in Excel to apply formatting to cells depending on the value contained in the cell. For instance, you can accentuate cells with values greater than a given figure or those within a given band.
- Data Bars, Colour Scales, and Icon Sets: These options give a graphical view of data. Data bars paint the cell with a gradient, colour scales use gradients for value distribution, and icons apply icons depending on the cell's value.

Formatting Data

1. Data Types:

- Text and Numbers: It's important that cells are formatted consistently as text or numbers. This can help avoid mistakes in calculations and analysis of the obtained data.
- Dates and Times: Excel has date and time as two separate data types. Format these using the 'Date' and 'Time' options in the Format Cells dialogue box.

2. Data Alignment and Orientation:

- Alignment: Justification makes your data a little more readable. To position data in cells, one can use horizontal and vertical alignment tools.
- Text Orientation: Rotating text can be done by using the 'Orientation' button located in the Home tab. There are more options when text is rotated, such as rotating it vertically or by an angle.

3. Formatting Tables:

- Creating Tables: To convert a range of data into a table, select your data, go to the Home tab, and then click Format as Table. This allows you to apply a consistent format that includes necessary table features such as filtering and sorting.
- Table Styles: When using Excel, there are different table styles that one can apply. These styles help format your table with headers, banded rows, and other features incorporated into the style.

4. Data Validation:

- Setting Rules: Data validation can be applied to regulate the kind of data that can be entered into a cell. For instance, you can set conditions that only characters between numbers or that dates should only be between a particular date range.
- Input Messages and Error Alerts: Use input messages to help users know what data to input and error messages to let users know that they have inputted invalid data.

5. Text Functions:

- Concatenation: Join text from two or more cells by using the CONCATENATE function or by using the ampersand (&) sign.
- Text Case: Use the functions UPPER, LOWER, and PROPER to change the case of a text to upper case, lower case, or proper case, respectively.

9.5 Sorting and Filtering of Data

Sorting and Subsorting Data in Microsoft Excel

- **Sorting Data**

1. Basic Sorting:

- Single Column Sort: To sort that data in a single column, first select any cell in the column you wish to sort. Then, click on the 'Data' tab available on the ribbon and choose either sort 'A to Z' for ascending order or sort 'Z to A' for descending order.
- Multi-Column Sort: When you want to sort the data by more than one column, you need to choose any cell on the data set. Then click on 'Sort' in the 'Data' tab. In the Sort dialogue box, it is necessary to introduce levels to define the columns further so as to sort and order them. For instance, you could first select the 'Last Name' followed by the 'First Name'.

2. Custom Sorting:

- Custom Lists: Once in a while, you may have to sort data in a certain manner in which you perceive the order, such as days of the week or months of the year. Here, in the Sort dialog box, you select 'Order' and then select 'Custom List.' This will allow you to sort according to your list.
- Case Sensitivity: To sort a list of case-sensitive items, go to the Sort dialogue box, check 'Options,' and check the 'Case-sensitive' box.

3. Sorting by Colour or Icon: Sorting by Colour or Icon:

- Cell Colour: If you have different cell colours in a column of your data, you can sort by colour. To do so, select the column you wish to sort by in the Sort dialogue box, select 'Cell Colour' under Sort On, and then select the order of the colour.
- Font Colour and Icons: Likewise, in the Sort dialogue box, you can choose 'Font Colour' or 'Cell Icon' to sort by the font colour or the cell icon.

- **Filtering Data**

1. Basic Filtering:

- Applying Filters: To filter, select any cell in your data range, go to the ‘Data’ tab, and click on ‘Filter.’ This will add drop-down arrows to every column header.
- Using Filters: As a result, the filter arrow will appear, and you need to click on the arrow located in the column you want to filter. It is then possible to check or uncheck the items in order to display or hide the rows which meet certain conditions. For instance, a filter is applied to a ‘Status’ column to display only such tasks which have a status of ‘Completed’.

2. Advanced Filtering:

- Text Filters: For the text data in the columns, use ‘Text Filters’ such as ‘contains’, ‘does not contain’, ‘begins with’, and ‘ends with’ for data view.
- Number Filters: To select subsets of numeric columns, use ‘Number Filters’ including ‘>’ or ‘<’ for ‘Greater Than’ or ‘Less Than,’ or ‘Between,’ or ‘Top 10.’
- Date Filters: For date columns, you can choose ‘Date Filters’ to filter data by certain or specific dates, a date range, and dynamic options such as ‘Last Month’, ‘This Year’, and ‘Next Week’.

3. Custom Filters:

- Custom Criteria: To select the additional filtering criteria, choose ‘Custom Filter’ from the filter drop-down. This enables one to use the logical operators ‘And’ or ‘Or’ to stipulate several conditions at the same time. For instance, you can select rows that meet the conditions of Sales being greater than 1000 and Region being East.

- **Knowledge Check 2**

  Fill in the Blanks

  1. Knowledge of ___________ and ___________ is important when it comes to data management in Excel. (functions & formulas)
  2. Log in and save ___________ is one of the preliminary things to do when using Excel. (sheet)
  3. Arithmetic ___________ and ___________ are primary prerequisites for computation in Excel. (operations & functions)
  4. ___________ enable users to control the look of cells in Excel and format data in the way they want. (Formatting options)

- **Outcome-Based Activity 2**

  Open Discussion and Comparison on Data Sorting and Filtering.

9.6 Summary

- Students quickly and comprehensively got the idea of workbooks and worksheets as basic tools for organising data in Excel, as well as their purpose and functions.
- We discussed opening and saving new workbooks, different file formats, and possible locations for saving workbooks.
- Familiarised with basic elements of Excel as well as simple arithmetic operations and 'simple' functions such as SUM, AVERAGE, etc.
- Cultural methods to format cells and data more effectively and attractively by using tools like font type, colour, borders, and alignment.
- Discussed and analysed approaches for arranging data in Excel with the help of the sort option to order the data in ascending, descending, or in a custom way utilising given criteria.
- Mastered one of the essential skills of structuring the data representation according to the set criteria, which allows for better data analysis and visualisation.
- Learned ways of inputting data and modifying data within Excel workbooks that the workbooks contain through the use of shortcuts and efficient ways.
- Familiarised with Excel's formula auditing tools, which help trace precedents and dependents, identify errors, and work with formulas efficiently.
- Investigated the data validation features in order to check the data validity and consistency where rules could be set and appropriate error messages to be displayed to the user.
- Understood elementary operations in worksheets, such as renaming, copying, moving, and deleting operations on worksheets, to manage Excel workbooks proficiently.

9.7 Keywords

- **Data manipulation:** Data manipulation is a process of altering, sorting, and analysing data to get necessary information for analysis or perform some operations.

- **Spreadsheet:** A spreadsheet is a document created using rows and columns that stores and processes data electronically.
- **Workbook:** A workbook in the context of Microsoft Excel is used to mean a book that is made of a number of worksheets where users of Microsoft Excel can store associated data in one file.

9.8 Self-Assessment Questions

1. How would you translate the concept of a workbook and a worksheet in MS Excel into understanding for a person who is a total beginner with the application?
2. Explain a case where formatting of cells and data helped in the presentation of the information in MS Excel.
3. Regarding the data sorting and filtering techniques in dealing with large data sets using MS Excel, explain the rationale behind it.

9.9 References / Reference Reading

- Bhavsar, Bharat S. *Mastering MS Excel 2022: Step-by-Step Guide for Beginners*. Tech Publications, 2022.
- Kumar, Ashok, and Rajeev Verma. *Microsoft Excel for Beginners: A Practical Guide for Indian Students*. New Age International Publishers, 2023.
- Lal, Vikas. *Excel for Everyone: A Step-by-Step Approach*. Pearson Education India, 2022.
- Walkenbach, John. *Excel 2023 Bible*. Wiley, 2023.

Unit 10: Advanced Features of MS Excel

Learning Outcomes:

- Students will be able to define advanced features of MS Excel.
- Students will be able to explain the use of advanced formulas and functions.
- Students will be able to create graphs and charts from data sets.
- Students will be able to apply data analysis tools like PivotTables.
- Students will be able to utilise conditional formatting and advanced charting techniques.

Structure:

10.1 Creating and Printing Workbooks/Sheets

Creating and Printing Workbooks and Sheets in Microsoft Excel

Creating Workbooks

1. Starting a New Workbook:

- Blank Workbook: To create a new workbook, open Excel and select 'Blank Workbook' on the start-up menu. If Excel is already running, go to the 'File' menu on the top left, hover over 'New,' and later click on 'Blank Workbook'.
- Templates: Users can find Excel templates for different uses, including budgets, calendars, and invoices. To use a template, click on the 'File' tab then 'New,' and you will be presented with the different templates you can select or even look for on the internet.

2. Adding and Managing Worksheets:

- Adding Worksheets: You can also right-click on the sheet tabs area and then choose the 'Insert' option. New worksheet symbols appear at the bottom of an Excel window as '+'. However, you can right-click on any of the sheet tabs and then click on 'Insert' and then 'Worksheet.'
- Renaming Worksheets: To rename a sheet tab, right-click on it and choose 'Rename', then enter a new name. You can also right-click on the sheet tab and choose a necessary name for it or double-click on it to provide a custom name.
- Reordering Worksheets: To move a sheet tab, click on it and then drag it to the required position within the workbook.
- Deleting Worksheets: On the left side of the Excel window, you right-click on the sheet's tab for the worksheet that you want to remove, then choose 'Delete. '

3. Organising and Formatting Worksheets:

- Colour Coding Tabs: To colour-code a sheet tab, right-click on it, choose 'Tab Colour' from the list, and pick any colour of your choice. This assists in giving the workbook a physical layout.
- Grouping Sheets: Right-click the selection of the sheets that you wish to group, and then click the Group and then the Ctrl key. This enables you to make changes on different sheets at the same time. To ungroup, simply right-click on any tab of the sheet that is in a group and choose 'Ungroup Sheets'.

Printing Workbooks and Sheets

1. Print Settings and Options:

- Accessing Print Options: Go to the menu at the top of the program, click on the 'File' option, then click on 'Print,' or use the 'Ctrl + P' keys on the keyboard to open the print settings.
- Print Preview: You can also notice the preview of your sheets depending on the settings you have made on the right side of the window. This helps you make corrections before going for the printout.

2. Page Setup:

- Margins and Orientation: In the print settings, to change the margins, simply click on 'Margins' and select from available standard options or set your own. There is also an orientation option between portrait and landscape.
- Scaling Options: There are 'Scaling' options to help you fit your worksheet onto one page or across a certain number of pages. This is useful for large datasets; big data, in this case, will have its separate folder.

3. Printing Specific Areas:

- Print Area: To print a particular cell or group of cells, one has to choose it, then click on the 'Page Layout' tab, click on 'Print Area,' and select 'Set Print Area.' This helps prevent another part of the worksheet from being printed.
- Print Titles: If your worksheet crosses over to multiple printed pages, one can have row or column headings appearing on each page. On the 'Page Layout' tab, go to the 'Print Titles' command and then enter the rows or columns that are to be repeated.

4. Headers and Footers:

- Adding Headers and Footers: To insert headers and footers, select the tab 'Insert' and find the option 'Header & Footer.' There, you can put in the desired text, page numbers, dates, etc.
- Customising Headers and Footers: On the 'Header & Footer Tools' Design tab, you can insert pre-defined or primary headers and footers and other options to add details like the file name or sheet name.

5. Printing Multiple Sheets:

- Selecting Sheets: When you want to print more than one sheet, you have to press the Ctrl button on the keyboard and click the desired tabs of the sheets. On the other hand, if the sheets are adjacent, select the first sheet tab, and while holding down the Shift key, click on the last sheet tab to select all the sheets between the two.

- Printing the Entire Workbook: Within the print settings, you can select print the entire workbook. This option prints all the sheets in the workbook in the order in which they are arranged.

10.2 Using Advanced Formulas and Functions

Using Advanced Formulas and Functions in Microsoft Excel

1. Array Formulas

Array Formulas:

- Basic Array Formula: Array functions can carry out one or several operations on one item or many items in an array. To generate an array formula, it is crucial to type the formula on the spreadsheet and press Ctrl + Shift + Enter simultaneously. For example, to sum a range of cells where certain conditions are met, you could use `{=SUM(IF(A1:In this case, the formula used is `=SUM(A10*5, B1:B10)`.

2. Logical Functions

IF, AND, OR:

- IF Function: The `IF` function performs logical comparisons for making decisions. For instance, `=IF(A1 > 10, "Yes", "No")` and the result of this function is “Yes” if the conditions of the function have been met in cell A1. Otherwise, it is “No”.
- AND Function: The `AND` function is used to determine if the conditions set are true or not. For instance, `=AND(A1 > 10, B1 < 5)` will give TRUE if both the conditions are fulfilled.
- OR Function: The `OR` function tests for any condition to be true out of the stated conditions. For example, `=OR(A1 > 10, B1 < 5)` will be TRUE if at least one of the conditions is true.

3. Lookup and Reference Functions

VLOOKUP, INDEX, MATCH:

- VLOOKUP Function: `VLOOKUP` is a type of formula that enables a user to look for a particular value in the first column of a table and then retrieve a value in the same row of a defined column. For example, `=VLOOKUP(A1, B1:`=VLOOKUP(A1, D10, 2, FALSE)` looks for the value in cell A1 in the range of cells B1 to D10 and pulls the value from the second column of the range.

- INDEX and MATCH: `INDEX` retrieves the value in the cell of a given row & column, whereas `MATCH` yields the relative position of a given value. For example, `=INDEX(B1:B10, MATCH(A1, A1:A10, 0)) took value from B1 to B10 where MATCH function pointed out the location of A1 in A1:A10.

4. Statistical Functions

SUMIFS, COUNTIES, AVERAGEIFS:

- SUMIFS Function: `SUMIFS` sums cells that meet multiple criteria. For example, `=SUMIFS(C1:C10, A1:A10, ">10", B1:B10, "<5")` sums the values in C1:C10 where A1:A10 is greater than 10 and B1:B10 is less than 5.
- COUNTIFS Function: `COUNTIFS` counts cells that meet multiple criteria. For example, `=COUNTIFS(A1:A10, ">10", B1:B10, "<5")` counts the cells in A1:A10 greater than 10 and B1:B10 less than 5.
- AVERAGEIFS Function: `AVERAGEIFS` calculates the average of cells that meet multiple criteria. For example, `=AVERAGEIFS(C1:C10, A1:A10, ">10", B1:B10, "<5")` averages the values in C1:C10 where A1:A10 is greater than 10 and B1:B10 is less than 5.

5. Text Functions

CONCATENATE, LEFT, RIGHT, MID, LEN, FIND:

- CONCATENATE Function: The `CONCATENATE` function joins multiple text strings into one. For example, `=CONCATENATE(A1, " ", B1)` combines the text in A1 and B1 with a space in between.
- LEFT, RIGHT, MID Functions: These functions extract parts of a text string. `=LEFT(A1, 3)` returns the first three characters of A1. `=RIGHT(A1, 3)` returns the last three characters. `=MID(A1, 2, 3)` returns three characters starting from the second character.
- LEN and FIND Functions: `LEN` returns the length of a text string, e.g., `=LEN(A1)` returns the number of characters in A1. `FIND` returns the position of a substring within a text string, e.g., `=FIND("text", A1)` returns the position of "text" within A1.

- **Knowledge Check 1**

  Fill in the Blanks

  1. Using blank space ___________ enables the generation and formatting of printable Excel data. (page layout)

2. Computer ___________ and memory make Excel capable of performing complicated arithmetic and computing data. (processing power)
3. Creating _______offers the means of creating pictures of the extracted data in Excel. (charts)
5. ___________ and ___________ enhance essential activities of analysing large and diverse data sets in Excel. (Filters & pivot tables)

- **Outcome-Based Activity 1**

  Explore Themes and Styles in MS -EXCEL.

10.3 Creating Graphs and Charts

Creating Graphs and Charts in Microsoft Excel

1. Understanding the Basics of Charts

Why Use Charts:

- Visual Representation: Data is presented graphically in charting, which makes it easier to find trends, varieties, and patterns, among other things.
- Data Interpretation: It makes complex data much easier to understand and interpret and enables faster decision-making processes.

2. The types of charts that are available in Excel are ;

Common Chart Types:

- Column and Bar Charts: These are especially useful when you need to analyze data within some subdivisions. Column charts are represented vertically, while bar charts are represented horizontally.
- Line Charts work well when demonstrating a change in values over time. They are suitable for showing scattered values that are united by a line.
- Pie Charts: These are most suitable for illustrating parts to a whole and other ratios. This applies where each slice on the pie chart corresponds to a certain category's value as a portion of the whole.
- Scatter Plots: Used to test theories involving cause & effect, difference & change, and difference between groups. Individual facts, numbers and information are plotted on an x-y plan.
- Area Charts: These are like line charts except that they are filled in down to the line. They are perfect for displaying aggregates over time, such as the total value of a portfolio over the years.

- Combination Charts: Overlay one or two data sets on the same chart or use two or more charts to display the data simultaneously. Rescaling serves to compare datasets that contain features in different scales.

3. Creating a Chart

1. Selecting Data:

- Highlight Data: Choose the range of values you need to reflect in a chart. Remember to add headers for improved identification.
- Including Multiple Series: To display multiple series, make sure your dataset is long enough and contains the categories and series to be put on the chart.

2. Inserting a Chart:

- Using the Ribbon: To do this, go to the ‘Insert’ tab in the Ribbon. Under ‘Charts,’ select the type of chart that you intend to design.
- Chart Recommendations: Next to the ‘Paste’ button, click on the drop-down icon and select ‘Recommended Charts’ to let Excel recommend a chart for your data.

3. Customising Chart Elements:

- Chart Title: Type the title in the empty box under the chart title. You can also format text in the box to change its font size and colour and align it properly.
- Axes Titles: To add titles to the axes, click on the chart and then the 'Chart Elements' button (a plus icon). Check 'Axis Titles' and enter your desired titles.
- Legend: Modify the legend by clicking on it and choosing its position or removing it if unnecessary.
- Data Labels: Add data labels by clicking 'Chart Elements' and checking 'Data Labels.' Customise their position and format for clarity.

10.4 Data Analysis Tools (PivotTables, Data Validation)

Data Analysis Tools in Microsoft Excel

1. PivotTables

Creating PivotTables:

- Selecting Data Range: Highlight the range of data you want to analyze. Include headers for better categorisation.
- Inserting a PivotTable: Go to the 'Insert' tab and click on 'PivotTable.' In the dialogue box, confirm the data range and choose where to place the PivotTable (new worksheet or existing one).

Building PivotTables:

- Adding Fields: Drag fields from the 'PivotTable Field List' into the 'Rows,' 'Columns,' 'Values,' and 'Filters' areas. For example, drag 'Product' to 'Rows' and 'Sales' to 'Values' to summarize sales data by product.
- Value Summarisation: Click on a field in the 'Values' area and choose 'Value Field Settings' to change how the data is summarised (e.g., Sum, Average, Count).

Customising PivotTables:

- Sorting and Filtering: Click the drop-down arrows in the PivotTable headers to sort or filter data. For advanced filtering, use 'Label Filters' or 'Value Filters'.
- Grouping Data: Right-click on a field and select 'Group' to group data. For example, you can group dates by month or quarter.
- Refreshing Data: If the source data changes, right-click the PivotTable and select 'Refresh' to update the analysis.

2. Data Validation

Setting Up Data Validation:

- Selecting Cells: Highlight the cells where you want to apply data validation.
- Opening Data Validation: Go to the 'Data' tab and click on 'Data Validation.'

Defining Validation Criteria:

- Allowing Specific Entries: In the 'Settings' tab, choose the type of data you want to allow (e.g., Whole Number, Decimal, List, Date).
- Creating Drop-Down Lists: Select 'List' in the 'Allow' box, then enter the values you want in the drop-down list, separated by commas, or reference a range of cells containing the list items.
- Setting Input Messages: The input Message tab allows you to set the message that appears when the user chooses the cell. This assists in data entry since they get to decide what data should be entered and what data should be excluded.
- Customising Error Alerts: In the ‘Error Alert’ tab, specify the message that is displayed when incorrect data is typed in. Select the type of alert to be used, which includes Stop Alert, Warning Alert, and Information Alert.

Using Formulas in Data Validation:

- Custom Validation: In the ‘Allow’ box, click on ‘Custom’, and then in the text box next to it, type in a formula to check the data. For example, `=ISNUMBER(A1)` checks that the entry is a number.

10.5 Conditional Formatting and Advanced Charting

Conditional Formatting in Microsoft Excel

Conditional formatting is one of the best features in Excel. Formatting is applied to cells depending on the specified conditions. Some of the ways that this can help include Underlining significant statistics that you cannot afford to overlook, drawing trends from the statistics, and Making it easier to read through the statistics.

1. Basics of Conditional Formatting

Purpose:

- Automatically set up the focus on the information, which is important according to the rules you set.
- They certainly help visualise data patterns and trends more easily.

Accessing Conditional Formatting:

- On the Ribbon, go to the ‘Home’ tab.
- Select the desired cell(s) to which conditional formatting should be applied, and click on ‘Conditional Formatting’ in the ‘Styles’ group.

2. Applying Basic Conditional Formatting

Highlight Cells Rules:

- Greater Than/Less Than Format cells that are above or below a certain value. For example, highlight cells greater than 100.
- Select the range of cells.
- Go to 'Conditional Formatting' > 'Highlight Cells Rules' > 'Greater Than...' and enter your value.
- Between Highlight cells that fall within a range. For example, between 50 and 150.
- Select the range of cells.
- Go to 'Conditional Formatting' > 'Highlight Cells Rules' > 'Between...' and enter the values.

Top/Bottom Rules:

- Top 10 Items: Highlight the top 10 (or any specified number) values.
- Select the range of cells.
- Go to 'Conditional Formatting' > 'Top/Bottom Rules' > 'Top 10 Items...'
- Bottom 10 Percent: Highlight the bottom 10 percent of values.
- Select the range of cells.
- Go to 'Conditional Formatting' > 'Top/Bottom Rules' > 'Bottom 10%...'

Colour Scales:

- Gradient Colouring: Apply a colour gradient based on cell values.
- Select the range of cells.
- Go to 'Conditional Formatting' > 'Colour Scales' and choose a colour scale.

Icon Sets:

- Icons Based on Values: Add icons to cells based on their values (e.g., traffic lights, arrows).
- Select the range of cells.
- Go to 'Conditional Formatting' > 'Icon Sets' and choose an icon set.

3. Creating Custom Rules

Using a Formula:

- Custom Conditions: As for more complicated conditions, it is possible to use the formation of rules based on custom formulas.
- Choose the cells to be included in the range.
- Choose ‘Home’ tab > ‘Styles’ group > ‘Conditional Formatting’ > ‘New Rule… ‘
- Select ‘Use a formula to determine which cells to format. ’
- To specify what you want to format (e.g., to highlight cells where the value is greater than 100, enter this formula: `=A1>100`).
- To set the formatting options, click the ‘Format. .. ‘ button.

Editing and Deleting Rules:

- Managing Rules: Modify the currently applied conditional formatting rules.
- Navigate to ‘Home’ > ‘Conditional Formatting ‘ > ‘Manage Rules’.
- Choose the rule to edit or delete and change it as you wish.

- **Advanced Charting in Microsoft Excel**

Other additional charting tools in Excel enable one to design better or more advanced charts on the data analysed. The following techniques can assist you in communicating insights and enhancing chart interactivity and elements of chart appeal.

1. Combination Charts

Creating Combination Charts:

- Purpose: Overlay two or more charts on a single chart in order to show two or more sets of data.
- Steps:
- Choose the data selection.
- Select the ‘Insert’ tab and then select ‘Combo Chart. ‘

- Choose the ‘Clustered Column – Line’ or another type of combo.
- Then, arrange each data series to employ different chart types, such as columns for one data series and lines for another series.

Customising Combination Charts:

- Secondary Axis: To further distinguish different ranges of data, it is time to insert a second axis.
- To change the data series to the secondary axis, right-click the desired data series.
- Choose ‘Format Data Series. ‘
- Select the check mark in ‘Secondary Axis’.
- Adjusting Chart Elements: These graphs clearly show that some of the titles, labels on the axes, and legends should be altered.

2. Dynamic Charts

Using Named Ranges:

- Dynamic Data Ranges: Set up the ranges by names that would change by themselves as data shifts.
- Click on the ribbon bar and navigate to ‘Formulas’, then ‘Name Manager’, and then choose ‘New’.
- Define a name and enter a formula that updates the range dynamically (e.g., `=OFFSET(Sheet1!$A$1,0,0, COUNTA(Sheet1!$A:$A),1)`).
- Use these named ranges in your chart data source.

Creating Dynamic Charts with Tables:

- Auto-Expanding Tables: Convert your data range into a table.
- Select the range and go to 'Insert' > 'Table.'
- Charts linked to this table will automatically update as data is added or removed.

3. Advanced Chart Types

Waterfall Charts:

- Purpose: Visualize the cumulative effect of sequential positive and negative values.
- Steps:
- Select your data range.
- Go to 'Insert' > 'Waterfall Chart.'
- Customize the chart to highlight different stages (e.g., starting point, intermediate steps, and ending point).

Box and Whisker Charts:

- Purpose: Display the distribution of data based on quartiles and identify outliers.

- Steps:
- Select your data range.
- Go to 'Insert' > 'Box and Whisker Chart.'
- Adjust the chart to show mean markers, quartile boxes, and whiskers.

Histograms:

- Purpose: Show the frequency distribution of a data set.
- Steps:
- Select the data range.
- Go to 'Insert' > 'Histogram Chart.'
- Customize bin ranges and intervals to represent your data distribution better.

4. Interactive Charts

Using Slicers with PivotCharts:

- Interactive Filters: Advanced slicers should be added to make your PivotCharts dynamic.
- Design a PivotTable and, in relation, a PivotChart.
- In the ‘Analyze’ tab, go to ‘Insert Slicer’ and then choose the fields that you want to act as the filter.
- To automatically change the representation of the PivotChart, use the slicers.

Using Form Controls:

- Interactive Elements: Insert buttons, checkboxes or drop-down lists to manipulate chart data.
- Choose ‘Developer’ > ‘Insert’ and choose a form control.
- Associate the form control with a cell.
- Change the chart data by using formulas or VBA in relation to the form control.

- **Knowledge Check 2**

  Fill in the Blanks

  1. ___________ and ___________ promote more data awareness and analysis to Excel. (Functions & formulas)
  2. The __________ feature of Excel helps in the data analysis and reporting of dynamic data. (PivotTable)
  3. ___________ in Excel help in identifying the patterns or even outliers of data. (Charts)

4. Specialisations allow for the performance of complex operations on the data in_______. (Excel)

- **Outcome-Based Activity 2**

  Open an MS Excel sheet, make 20 entries of roll no. and name of the student, and Freeze 1st row.

10.6 Summary

- Students understood the most effective techniques for working with the workbooks and sheets in Excel.
- Discussed the application of advanced forms and functions that are used for more practical purposes, such as performing calculations and analysis.
- We have covered the techniques to have effective features in graphs and charts by mastering them to have an effective presentation of data.
- Learned about the Data Analysis Tools, Applied PivotTables feature, and Data Validation in order to analyse and validate data.
- Acquired knowledge on how to use conditional formatting in order to make important patterns and trends stand out.
- Understood formatting workbooks and sheets, as well as printing them in the best look possible.
- Learned Data Validation, which measures to check the accuracy of the data entered in the database.
- Mastered the advanced Excel functions like VLOOKUP, INDEX, MATCH, etc. The strategies for sorting and handling big workbooks were also explored.

10.7 Keywords

- **Data manipulation:** Data manipulation involves the act of changing, sorting and analysing data in order to get insights or for the purpose of accomplishing certain tasks**.**
- **Spreadsheet:** A spreadsheet is a type of document with rows and columns where users can input and process data.
- **Workbook:** In Microsoft Excel, a workbook is a file made of several sheets or worksheets that organise related information in a single file.

10.8 Self-Assessment Questions

1. Reflect on your experience with creating and printing workbooks/sheets in MS Excel. What challenges did you encounter, and how did you overcome them?
2. Explain why the application of MS Excel, which involves the utilisation of various formulas and functions in order to perform complicated calculations or data manipulation, is important.
3. Describe a graph or chart you have made using MS Excel. In what way was it more effective in presenting your data than if you converted it into tables?
4. Discuss the advantages of PivotTables and Data Validation in the process of data analysis in MS Excel. Could you give a real-life example of their use?

10.9 References / Reference Reading

- McCartney, Mike. *Microsoft Excel 2023: The Most Updated Crash Course from Beginner to Advanced*. Barnes & Noble, 2023. ISBN: 9798868928253.
- Roberts, Alan. *EXCEL 2023: The Practical Step-by-Step Manual of Microsoft Excel for Learning Basic and Advanced Features, Formulas, and Charts with Easy and Clear Examples*. Amazon, 2023.
- Wilson, Richard. *EXCEL 2023: From Beginner to Expert | The Illustrative Guide to Master All The Essential Functions and Formulas in Just 7 Days With Step-by-Step Tutorials, Practical Examples, Tips & Tricks*. Amazon, 2023.
- Jain, Rupesh K., and Sharma, Nikhil. *Mastering MS Excel 2023: Advanced Techniques and Data Analysis Tools*. New Delhi: Tech Books International, 2023.
- Kapoor, Anil. *Advanced Excel 2023: A Comprehensive Guide for Professionals*. Mumbai: Excel Publishing House, 2023

Unit 11: Introduction to MS PowerPoint

Learning Outcomes:

- Students will be able to identify key features of MS PowerPoint.
- Students will be able to explain how to create and save presentations.
- Students will be able to demonstrate how to insert and format text and images.
- Students will be able to apply slide layouts and themes.
- Students will be able to utilise animations and transitions effectively.

Structure:

11.1 Creating Presentations

Microsoft PowerPoint is well-known, and creating a presentation is a very important skill in presenting information. Here is a step-by-step guide on how to create a polished and professional presentation in PowerPoint:

Starting a New Presentation

Opening PowerPoint

- Launch PowerPoint: Launch the application either from your desktop or the start menu.
- New Presentation: Click on the “New” button when on the start window to start a new presentation from a blank slate or use a template.

Choosing a Template

- Templates: PowerPoint has numerous templates with already-created designs for the slides. To ensure that you select the right template, choose a theme that relates to your presentation.
- Blank Presentation: As a result, to produce a professional and unique presentation, the user starts with an empty slide and designs each slide independently.

Adding and Formatting Slides

Adding Slides

- New Slide: To add a new slide, go to the toolbar, select “Home,” and click “New Slide.” You can choose from many different layouts, including the title slide, the content slide, or the section header.
- Layout: You can reorganise an existing slide by right-clicking on it, clicking on the ‘Home’ tab, selecting ‘Layout,’ and selecting the desired layout.

Formatting Slides

- Background: To adjust the slide's background, go to ‘Design’ and then ‘Format Background,’ where you can set a solid fill colour, gradient fill, picture fill, or texture fill.
- Themes: Choose a theme in the “Design” tab to make the presentation look more uniform.

Adding Content

Text

- Text Boxes: To place text boxes on the slides, go to the toolbar and click “insert,” then click on “text box”.

- Formatting Text: Under the Home tab, use the available options for formatting the text, which are font type, size, colour, and alignment.

Images and Media

- Images: To embed images, go to the ‘Insert’ menu and click ‘Pictures,’ then select from your documents. Shrink and align them appropriately if they do not fit the space provided.
- Online Pictures: To locate images using the search within PowerPoint, go to the “Insert” tab and click on “Online Pictures.”
- Videos: Insert videos by going to the tab “Insert” and clicking on “Video. “ You can upload videos from your computer or even link to the videos that you want.
- Audio: Add recording by going to “Insert” > “Audio” to add music or recording.

11.2 Inserting and Formatting Text and Images

Inserting and formatting text and images in Microsoft PowerPoint are essential skills for creating engaging and visually appealing presentations. Here’s a detailed guide to help you master these tasks:

Inserting Text

Applying text and images in the context of Microsoft PowerPoint is one of the crucial activities for making captivating and appealing presentations. Here’s a detailed guide to help you master these tasks: Here’s a detailed guide to help you master these tasks:

Inserting Text

Adding Text Boxes

1. New Slide: You can add a new slide if it is required. It is located at the top of the PowerPoint window; first, go to the “Home” tab and then click the “New Slide” button.
2. Text Box: To add a text box, go to the “Insert” tab of the ribbon and click on the button labelled “Text Box.”
3. Draw Text Box: On the slide, use the cursor to draw the text box to the preferred size and shape.

Formatting Text

1. Font Style: Right-click the text and pick a font style from the “Home” tab of the toolbar.
2. Font Size: Modify the text from the “Change the Font Size” drop list.

3. Font Colour: To erase font colour, click on the “Font Colour” button located in the “Font” tab.
4. Bold, Italic, Underline: Use the buttons in the “Home” tab or hotkeys (Ctrl + B, Ctrl + I, Ctrl + U) with the names of these styles.
5. Alignment: The icons in the “paragraph” tab allow one to left-align, centre, right-align, or justify the text.
6. Bullets and Numbering: Incorporate bullets or numbering into the lists by the icons in the “Paragraph” tab.

Inserting Images: Import/Insert Pictures from your Computer

1. Insert Image: Head over to the “Insert” tab, then click on “Pictures.”
2. Select Image: For the Output settings, select “This Device” and navigate through the folders in your computer to pick the image file.
3. Insert: When you are ready to place the image in your slide, go to the top of the page and click on “Insert.”

Adding Online Images

1. Insert Online Picture: Next, go to the “Insert” tab and click on “Pictures.”
2. Online Pictures: Select “Online Pictures” and then type in the picture that is required.
3. Insert: Click on the picture you want, and then click on “Insert.”

Formatting Images

Resizing Images

1. Select Image: To select an image, click on it.
2. Resize Handles: Click on one of the corner handles of the image and then drag it to make the image larger or smaller, but adjust it in proportion. The side handles can be used to pull the image in or out.

Positioning Images

1. Drag and Drop: To do this, click and drag the image to the appropriate position on the slide you wish to use.
2. Align The other tools to help position the image are available under the tab known as “Picture Format.”

Adding Effects to Images

1. Picture Styles: The “Picture Format” tab offers many pre-set styles.
2. Picture Border: A border can be added by going to “Picture Border” and then choosing each colour, weight and dash style.

3. Picture Effects: From the “Picture Effects” menu, add effects like shadow, reflection, glow, etc on the picture.

- **Knowledge Check 1**

  Fill in the Blanks

  1. Designing a presentation in the PowerPoint involves the development of ___________. (slides)
  2. Applying ___________ and alignment between and around the text also makes the PowerPoint slides more appealing. (spacing)
  3. This means that when a user inserts slide ___________, this will help create uniformity in designing the PowerPoint. (templates)
  4. Introducing ___________ and ___________ between slides increases the interactivity of presentations created using PowerPoint. (transitions & animations)

- **Outcome-Based Activity 1**

  Prepare a slide show and attempt to add photos.

11.3 Using Slide Layouts and Themes

Applying the slide layouts and themes in Microsoft PowerPoint makes it easier to create uniform and attractive presentations. Here's a detailed guide on how to effectively use slide layouts and themes:

Using Slide Layouts

Applying a Slide Layout

1. New Slide: To add a new slide to the presentation, in the “Home” tab, select “New Slide” in the dropdown list.
2. Changing Layout: To modify the arrangement of the content in one of the existing slides, first, we have to select the slide we want to modify, after which we go to the “Home” tab and then click the “Layout” button which is on the “Insert” group, and we select the layout of our choice from the dropdown list.

Common Slide Layouts

1. Title Slide: Applied for the title of the presentation and the subtitle of the presentation.

2. Title and Content: This section consists of a title bar and a text bar for the report's content, including images, charts, and any other additional content.
3. Section Header: Employed to bring in a new section, a title, and a subtitle of the news section.
4. Two Content: Contains two content boxes, one beside the other.
5. Comparison: Like Two Content, but there are titles for each content box.
6. Title Only: A slide that has only a title box on it.
7. Blank: A slide which does not have any placeholders before the speaker sets the type.

- **Using Themes**

Applying a Theme

1. Choose a Theme: Navigate to the “Design” tab of the program. In the Themes group, you will find different themes that will be set in the online Journal.
2. Apply Theme: Click on a theme to select it for your whole presentation. Roll over a theme to see the preview in action on your slides.
3. More Themes: The Themes list continues after the Themes group; to open it, click the More button, which looks like a downward arrow.

Customising Themes

1. Colours: You can change the theme colours by going to “Design” > “Variant” > “Colours.” You can choose a preset colour or design a new one.
2. Fonts: Alter the theme fonts by going to the ‘Design’ tab, followed by ‘Variants’ and then ‘Fonts.’ Select a font set or use a custom one.
3. Effects: Modify it by going to “Design, " then ‘Variants,’ and finally ‘Effects.’ This alters the look of features such as shadows and reflections.
4. Background Styles: You can change the background by going to “Design” > “Variants” > “Background Styles. ” Here, you can choose from ready-made styles or new ones.

11.4 Adding Animations and Transitions

- **Ribbon Menu system in MS PowerPoint: Animations Tab**

The “Animations” tab of the Ribbon Menu system is one of the essential parts of the MS PowerPoint application, and it contains a range of tools and settings to create an animated slide show. Here is a detailed description of the features available in the "Animations" tab:

Animation Gallery: The Animation Gallery presents a large number of pre-programmed animation schemes that can be used to animate one or several objects on the selected slide. To add a bit of life, users can choose from entrance, emphasis, exit, and motion paths to enhance their content. The gallery offers a range of beautiful and properly created animations for various modes of presentations.

Animation Options: This section enables users to adjust the selected animating effect to their desired style. The settings of the animation can also be edited by the users in relation to direction, time, and even duration, among other things. They can also define whether the animation loads when the page does or when the user clicks on something.

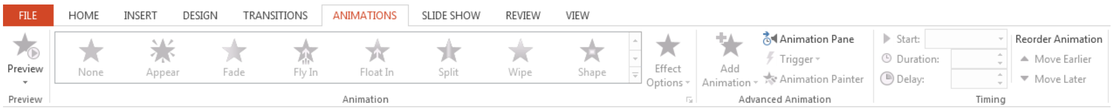


Fig: Ribbon Menu system in MS PowerPoint: Animations Tab

Animation Timing: The animation timing options allow one to set the time for the sequence and the time that an animation takes on the slide. One can set how long after one animation, another should occur, whether one animation should occur before another or at the same time, and how fast the animations occur. This feature makes the various actions undertaken in the presentation appear flowing and in harmony with the others.

Animation Preview: The animation preview option enables the user to show how the animation would look on the slide. The 'preview' button allows users to view a real-time preview of an animation with the intention of enhancing its look before presenting it.

The 'Animations' tab in PowerPoint helps users create proper and appealing animations for presentations. The provided tools and features attract the audience and explain the main idea properly.

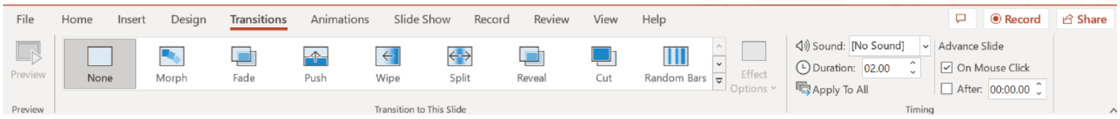
- **Ribbon Menu system in MS PowerPoint: Transitions:**

In Microsoft PowerPoint, "Transitions" is one of the essential tools situated in the Ribbon Menu system, which enables users to make changes to the ways the slides transition in a presentation. Here is a comprehensive description of the features available in the "Transitions" tab: Here is a comprehensive description of the features available in the "Transitions" tab:

Transition to This Slide: This section offers a list of transition effects that may be assigned on a slide-by-slide basis. Transition effects relate to the appearance and disappearance of the slides during a presentation and make the presentation more attractive and fluent. Some of the options available are fades, wipes, blinds, and so on.

Timing Group: The timing group enables the setting of the length of each slide and the speed of the transition effect. Users can decide how long a slide is to be displayed before the next one is displayed. They can also decide the speed and how smoothly the transition effect will work through the selected transition duration.

Sound Group: The sound group contains options for adding sound to the slides' transitions. Transition sound can be selected from the given library or uploaded as an audio file. The addition of sounds can extend the benefits of the visual change and attract the audience's attention to the audio cue.



Sound: [No Sound]
Duration: 02.00
☑ On Mouse Click
□ After: 00:00.00

Fig.: Ribbon Menu system in MS PowerPoint: Transitions Tab

Preview Group: The preview group supports a preview pane that helps depict the live preview of the selected transition and the time over which the transition will be done. Users can tailor specifics and immediately get a look at how the transitions will look when executed in the real presentation. This feature assists the user to adjust the effects to the level that the user wants it to be.

Effect Options: The effect options section is used when applying certain transition effects to provide additional options for the transition. In addition to this, users can select specific variations or variations with sound for each transition effect that is likely to be applied, offering users more customisation options and more control over the transitions being made.

Apply to All: The “Apply to All” button enables a user to apply the chosen transition effect on all the slides in the specific presentation. This feature is convenient when you need to use the same transition effect for all the slides, creating a more professional-looking presentation.

11.5 Basics of Slide Design

- **Ribbon Menu system in MS PowerPoint: Design Tab:**

The "Design" tab in Microsoft PowerPoint is an essential component of the Ribbon Menu system that allows users to customize the appearance and layout of their presentations. Here is a comprehensive overview of the features available in the "Design" tab:

- **Themes Group:** The themes group offers you a list of professionally developed themes, which give a presentable outlook to your presentation. Most themes incorporate colours, fonts and effects that will ensure the entire appearance of the slides is harmonised and professional.

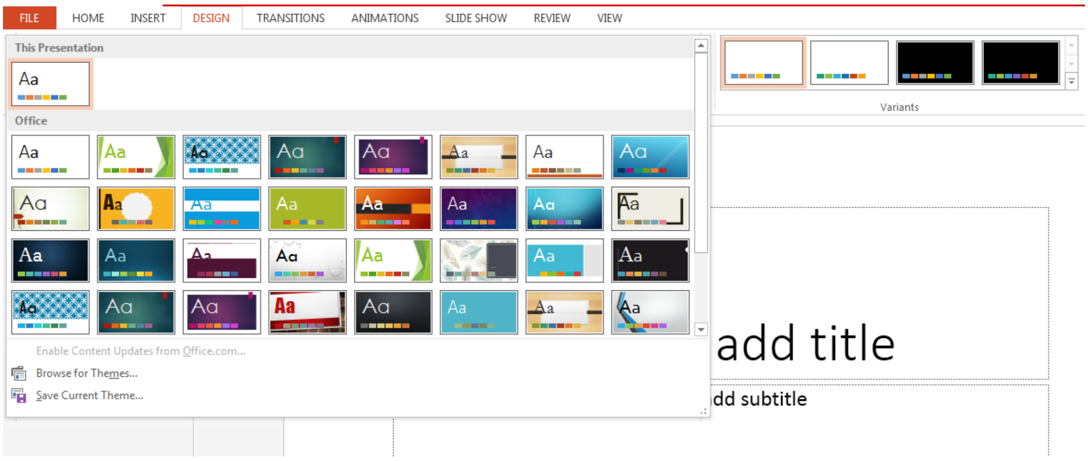


Fig: Ribbon Menu system in MS PowerPoint: Design Tab

- **Slide Size Group:** The slide size group allows you to change the size and orientation of your slides. Slides come in standard sizes, or you can order the size you require based on your presentation’s requirements. This feature is helpful in preparing presentations for various devices or resolutions.
- **Customize Group:** The customised group has more choices concerning the customisation of the slide background. The background can be any solid colour, gradient, picture, or pattern that you want. This enables you to make the background attractive and correspond to your presentation's content.
- **Slide Layout Group:** This group enables you to choose from several slide designs already provided for you. Slide designs define the position of content in each slide, including text, images, charts, and other objects. There are layout options that you can select, and you can easily change the layout depending on your content.
- **Themes Options Group:** It is a subpanel of the theme that offers additional settings for the selected theme among various groups of options. This aspect enables you to change the theme colours, fonts and effects to meet your specific preference for the

theme of the presentation. This feature allows you to add the finishing touches to the visual aspects of your slides.

- **Knowledge Check 2**

  Fill in the Blanks

  1. Fundamentals of __________ govern the general look and consistency of the presentations created with PowerPoint. (design)
  2. ___________ enable changes and restructuring of the content of PowerPoint slides. (Slide layouts)
  3. Using PowerPoint, one can create __________ and also set the background and theme for the slides. (slides)
  4. Adding ___________ and ___________ makes any PowerPoint presentation more interesting and informative. (images & charts)

- **Outcome-Based Activity 2**

  Work on the Basics of Slide Design and create 10 slides**.**

11.6 Summary

- Microsoft PowerPoint is well-known, and creating a presentation is a very important skill in presenting information. We have seen the Presentation Creation in MS PowerPoint, which is used to develop coherent and visually informative lessons.
- Inserting and formatting text and images in Microsoft PowerPoint are essential skills for creating engaging and visually appealing presentations. Acquired the ability to place text and images into the document and make them look presentable.
- Learned different slide layouts and themes for the continuity and aesthetically pleasing appearance of the presentation.
- The “Animations” tab of the Ribbon Menu system is one of the essential parts of the MS PowerPoint application. It contains a range of tools and settings to create an animated slide show. Animations and transitions improve the flow and interaction of presentations.
- Learned about key concepts to use when designing a slide so that your message can be well understood.

- Identified the techniques of engaging the audience at the beginning of the presentation and throughout the presentation.

11.7 Keywords

- **Animations:** In MS PowerPoint, animations are special effects applied to the text, objects, or the whole slide, making the presentation more impressive and full of movements and transitions.
- **Multimedia Elements:** Media objects in MS PowerPoint are features of media like images, videos, audio files, and animations that are used in the presentation to help inform the audience more livelyly.
- **Slide Show:** A slide show in MS PowerPoint is a set of slides shown one after another in order. It describes a story or provides information to an audience and uses animations, multimedia objects, and other transitional effects between the slides.

11.8 Self-Assessment Questions

4. What factors do you consider when designing a presentation in MS PowerPoint?
5. Explain how you would go about inserting text and images and also formatting these text and images in the MS PowerPoint Slides.
6. Explain the significance of using templates and design in communicating information in an MS PowerPoint presentation.
7. Where and how should one apply animations and transitions in an MS PowerPoint presentation?
5. Discuss some guidelines that you consider fundamental when designing slides in order to come up with an aesthetically appealing and comprehensible presentation in MS PowerPoint.

11.9 References / Reference Reading

- Agarwal, Saurabh. *Mastering Microsoft PowerPoint: A Comprehensive Guide for Students and Professionals*. BPB Publications, 2023.
- Bhattacharya, Priti. *Microsoft PowerPoint 2023 for Beginners: Step-by-Step Guide to Creating Professional Presentations*. TechKnow Books, 2023.
- Joshi, Ramesh, and Mehta, Kavita. *Effective Presentation with Microsoft PowerPoint: Techniques and Tips*. Wiley India, 2022.

- Reding, Elizabeth Eisner. *Microsoft PowerPoint 365 Complete: In Practice*. Cengage Learning, 2023.
- Walkenbach, John, et al. *Microsoft PowerPoint 2023 Bible*. Wiley, 2023.

Unit 12: Advanced Features of MS PowerPoint

Learning Outcomes:

- Students will be able to define the purpose of master slides.
- Students will be able to explain how to incorporate multimedia elements.
- Students will be able to demonstrate advanced animation techniques.
- Students will be able to apply methods for delivering and printing presentations.
- Students will be able to create interactive presentations using advanced features.

Structure:

12.1 Creating Master Slides

Creating Master Slides in Microsoft PowerPoint

Master Slides in PowerPoint enable you to keep a consistent outlook of your presentation in terms of the design, layout, and formatting of the presentation slides. This guide will take you through all the steps to create and apply Master Slides in a clear and structured manner.

1. Understanding Master Slides

What Are Master Slides?

- Slide Master: The highest level of organisation of the slide layout in which general settings for all the slides are made.
- Layout Masters: Reporting to the Slide Master, these govern the format of various kinds of slides (for instance, Title Slide, Content Slide).

Why Use Master Slides?

- Consistency: The presentation should be presented as one set, and all the slides should have a similar layout.
- Efficiency: Make changes that affect the whole presentation quickly by using the Master Slides.
- Customisation: Make and use specific pre-designs that can be applied to the field in a pro forma manner.

2. Customising the Slide Master

Editing the Slide Master:

- Background: The next one is to set a background for your slides.
- Since all the formatting options for the presentation are available in the Slide Master view, the next step is to ensure that it is activated.
- Click on the ‘Slide Master’ tab and select ‘Background Styles’. You can then select the desired background or create a new one.
- Fonts: Use appropriate colours and sizes of the font.
- go to the slide Master’ tab, click on ‘Fonts’ and either choose a font theme or create a font scheme of your choice.
- Colours: Once you have decided on the kind of pool you want, you should select the best colour for your pool.
- Navigate to the ‘Slide Master’ tab and then click on ‘Colours,’ and then select a theme or set colour.

- Effects: Place shadows or reflections that share the look of other objects in the scene.
- Choose an effect theme for the presentation by clicking on ‘Effects’ located in the ‘Slide Master’ tab.

Adding and Formatting Placeholders:

- Types of Placeholders: The tools bar permits you to insert elements such as plain text, pictures, charts, tables, and others.
- Inserting Placeholders: To add a placeholder, go to the Slide Master or Layout Master view, click on the Insert Placeholder tab in the Slide Master section and select the desired type of placeholder.
- Formatting Placeholders: This means adjusting the size of the placeholders, changing their position on the page, and/or modifying the grid template they are placed into. Select a placeholder and then go to the ‘Home’ tab, where you will find the formatting tools.

3. Creating Custom Layouts

Adding a New Layout:

- To get to the Slide Master view, go to the ‘Slide Master’ tab at the top of the PowerPoint application interface.
- To add this or any other layout, they can click on the ‘Insert Layout’ button.

Customising Layouts:

- Adding Elements: You can add placeholders, text boxes, images, and shapes to the layout that you will customise.
- Formatting Layouts: I will add the background styles, fonts, and colours so that the layout is in harmony with the Slide Master.
- Renaming Layouts: On the left pane, right-click on the new layout created and click on ‘Rename Layout. ‘Ensure it has a good name for identification.

4. Master Slides and Their Use in Making Your Presentation

Using Layouts in Normal View: Using Layouts in Normal View:

- Applying a Layout: In Normal view for PowerPoint, select the ‘Home’ tab, click on ‘Layout,’ and select layouts from those designed in the Slide Master.
- Changing Layouts: You can also change the layout of any slide by clicking on it and then selecting ‘Layout’ from the drop-down menu.

Updating Slides with Master Changes: Updating Slides with Master Changes:

- Any modification done on the master Slide or Layout Masters will affect all slides that have been built on such formats in your presentation.

12.2 Incorporating Multimedia (Audio, Video)

Incorporating Multimedia in Microsoft PowerPoint

Adding multimedia elements like audio and video can enhance your PowerPoint presentations and engage your audience on a deeper level. This guide will walk you through the process of incorporating audio and video seamlessly into your presentations.

1. Inserting Audio

Adding Background Music:

- Click on the slide to which you wish to add the audio.
- Go to the ‘Insert’ tab of the Ribbon.
- To display the audio options, go to the ‘Media’ tab and left-click on the option that says ‘Audio’.
- Select ‘Audio on My PC’ to insert audio files located on your computer.
- Click on the audio file you want to use and then click on ‘Insert. ‘

Customising Audio Playback:

- Click on the audio icon on the slide.
- Go to the 'Playback' tab on the Ribbon to access options like 'Start Automatically' or 'Loop Until Stopped.'
- Adjust volume levels and set fade-in or fade-out effects as needed.

Adding Narration or Voiceover:

- Go to the 'Insert' tab and click on 'Audio.'
- Choose 'Record Audio' to record narration directly into PowerPoint.
- Speak into your microphone and click 'OK' when finished recording.
- Adjust playback options and save the narration.

2. Inserting Video

Embedding Video from File:

- Navigate to the slide where you want to insert the video.
- Go to the 'Insert' tab on the Ribbon.
- Click on 'Video' in the 'Media' group.
- Choose 'Video on My PC' to insert video files saved on your computer.
- Select the video file and click 'Insert.'

Inserting Online Videos:

- Click on 'Online Video' instead of 'Video on My PC.'
- Enter the URL of the online video or search for it using Bing Video Search.
- Select the video and click 'Insert.'

Adjusting Video Playback Options:

- Click on the video on the slide.
- Go to the 'Playback' tab on the Ribbon to access options like 'Start Automatically' or 'Rewind After Playing.'
- Set up additional options like volume control and video poster frame.

Using Video Triggers:

- Create interactive presentations by using video triggers.
- Insert a shape or object on the slide.
- Assign the action to 'Play Media' and select the video you want to play when the shape or object is clicked.

- **Knowledge Check 1**

  Fill in the Blanks

  1. Creating ___________ ensures consistency in formatting across all PowerPoint slides. (slide master templates)
  2. Incorporating ___________ and ___________ enriches PowerPoint presentations with multimedia elements. (images & videos)
  3. Advanced ___________ techniques add dynamism and sophistication to PowerPoint presentations. (animation)
  4. Delivering and ___________ presentations effectively communicate ideas and information in PowerPoint. (presenting)

- **Outcome-Based Activity 1**

  Create 5 slides on your university.

12.3 Advanced Animation Techniques

Advanced Animation Techniques in Microsoft PowerPoint

Advanced animation techniques in PowerPoint allow you to create dynamic and engaging presentations that capture your audience's attention. From complex motion paths to interactive triggers, these techniques will help you take your presentations to the next level.

1. Custom Motion Paths

Creating Custom Motion Paths:

- Select the object you want to animate.
- Go to the 'Animations' tab on the Ribbon.
- Click on 'Add Animation' and choose 'More Motion Paths' from the drop-down menu.
- Select the desired motion path from the options or draw a custom path using the 'Custom Path' tool.

Editing Motion Paths:

- After applying a motion path, click on the object to reveal the motion path.
- Drag the green handles to adjust the shape of the path.
- Use the yellow diamond to adjust the direction of the path.

Combining Motion Paths:

- Apply multiple motion paths to the same object to create complex animations.
- Use the 'Start' drop-down menu in the 'Animations' tab to sequence the motion paths.

2. Morph Transition

Using the Morph Transition:

- Create two slides with objects you want to animate.
- Ensure the objects are in the same position and have the same size on both slides.
- Go to the 'Transitions' tab on the Ribbon.
- Select 'Morph' from the transition options.

Customising the Morph Transition:

- Click on the 'Effect Options' button in the 'Transitions' tab.
- Adjust the duration and other settings for the Morph transition.

Creating Complex Animations with Morph:

- Use Morph to animate multiple objects, text, and shapes across slides.
- Experiment with resising, rotating, and moving objects to create dynamic effects.

3. Triggered Animations

Adding Triggers:

- Select the object you want to animate.
- Go to the 'Animations' tab on the Ribbon.

- Click on 'Add Animation' and choose an animation effect.
- Click on 'Trigger' in the 'Advanced Animation' group and select 'On Click of' or 'On Click of Object.'

Using Shapes as Triggers:

- Insert a shape (e.g., a button) on your slide.
- Apply the 'No Fill' and 'No Outline' formatting to make the shape invisible.
- Assign the shape as a trigger for specific animations.

Creating Interactive Presentations:

- Use triggers to create interactive presentations where user actions activate animations.
- Combine triggers with hyperlinks to navigate between slides or external content.

12.4 Delivering and Printing Presentations

Delivering and printing presentations in PowerPoint involves more than just creating slides. From rehearsing timings to choosing the right print settings, this guide will help you prepare and present your slides effectively, whether in person or print.

1. Delivering Presentations

Rehearsing Timings:

- Go to the 'Slide Show' tab on the Ribbon.
- Click on 'Rehearse Timings' in the 'Set Up' group.
- Practice your presentation, and PowerPoint will record the time spent on each slide.

Setting up Slide Show Options:

- Click on 'Set Up Slide Show' in the 'Set Up' group.
- Choose options like 'Show Type' (e.g., Presented by a speaker, Browsed by an individual) and 'Show Options' (e.g., Loop continuously until 'Esc').

Presenting Slides:

- Go to the 'Slide Show' tab on the Ribbon.
- Click on 'From Beginning' or 'From Current Slide' in the 'Start Slide Show' group to start presenting.

Using Presenter View:

- If using dual monitors, enable Presenter View for additional control and notes.
- Go to the 'Slide Show' tab and click on 'Presenter View' in the 'Monitors' group.

2. Printing Presentations

Printing Slides:

- Go to the 'File' tab on the Ribbon.
- Click on 'Print' in the left pane.
- Choose print settings like the number of copies, colour or grayscale, and slide range.
- Click 'Print' to send the presentation to the printer.

Printing Speaker Notes:

- In the 'Print' settings, select 'Notes Pages' under 'Settings.'
- This option prints one slide per page with accompanying speaker notes below each slide.

Printing Handouts:

- Choose 'Handouts' under 'Settings' in the Print settings.
- Select the desired number of slides per page (e.g., 1, 2, 3, 4, 6).
- Customize other settings like colour or grayscale and slide order.

Printing Outline View:

- If you prefer a text-based outline of your presentation, choose 'Outline View' under 'Settings.'
- This option prints the titles and text from each slide in outline format.

12.5 Interactive Presentations

Interactive presentations engage your audience and encourage participation, making your message more memorable and impactful. PowerPoint provides interactivity, where you can include quizzes, navigation bars, and other elements that allow viewers to participate. This guide will lead you through the steps of designing an engaging presentation with your audience in mind.

1. Hyperlinks

Creating Hyperlinks:

- To create a hyperlink, choose the object you want to link. It can be text, a shape, or an image.
  - Go to the tools bar at the top, then right-click the selected text and click on the hyperlink.
- In the 'Insert Hyperlink' dialogue box, choose the destination for the hyperlink (e.g., another slide, a website, an email address).
- Click 'OK' to apply the hyperlink.

Types of Hyperlinks:

- Slide Hyperlinks: Link to specific slides within the presentation for seamless navigation.
- Webpage Hyperlinks: Direct viewers to external websites for additional information.
- Email Hyperlinks: Allow viewers to send emails directly from the presentation.

Creating Navigation Menus:

- Design a menu slide with clickable buttons for different sections or topics.
- Link each button to the corresponding slide using hyperlinks.
- Add a 'Back to Menu' hyperlink on each slide to return to the menu slide.

2. Action Buttons

Inserting Action Buttons:

- Go to the 'Insert' tab on the Ribbon.
- Click on 'Shapes' in the 'Illustrations' group.
- Choose an action button shape (e.g., Home, Next, Previous, Information).
- Click and drag to draw the button on your slide.

Assigning Actions to Buttons:

- Right-click on the action button and choose 'Action Settings.'
- In the 'Action Settings' dialogue box, select the action you want the button to perform (e.g., hyperlink to a slide, run a program, play a sound).
- Click 'OK' to apply the action.

Creating Interactive Navigation:

- Use action buttons as navigation controls for moving between slides.
- Customize action buttons with text, icons, or colours to indicate their function.

3. Interactive Quizzes

Creating Quiz Questions:

- Design quiz slides with multiple-choice questions, true/false statements, or fill-in-the-blank prompts.
- Use text boxes, shapes, or SmartArt to create answer choices.

Adding Hyperlinks to Answers:

- Link each answer choice to a feedback slide using hyperlinks.
- Create feedback slides with explanations or responses based on the user's answer selection.

Using Slide Zoom for Answer Reveals:

- Insert slide zooms on the feedback slides to zoom in on the correct answer or provide additional information.
- Link the zoomed-in area back to the main quiz slide to allow users to continue the quiz.

4. Interactive Elements

Inserting Interactive Objects:

- Use features like sliders, spinners, and checkboxes from the 'Developer' tab (enable it in PowerPoint options if not visible) to create interactive elements.
- Link these interactive objects to specific actions or animations using macros (VBA programming).

Creating Interactive Games:

- Design games like Jeopardy, trivia, or board games using interactive elements and hyperlinks.
- Incorporate animations, sound effects, and scoring systems to enhance the gaming experience.

Using Interactive Templates:

- Explore interactive PowerPoint templates available online or create your templates with pre-designed interactive elements.
- Customize templates to match your presentation content and style.

- **Knowledge Check 2**

  Fill in the Blanks

  1. ___________ presentations motivate the audiences and have them actively involved in the use of PowerPoint. (Engaging)
  2. A __________ feature of PowerPoint is the ability to incorporate audio and video files in the presentation. (multimedia)
  3. ___________ allow the creation of animal sequences and impact in PowerPoint. (Transitions)
  4. Applying ___________ helps in easy navigation from one slide to another in PowerPoint. (hyperlinks)

- **Outcome-Based Activity 2**

  Create / Include multimedia objects in a presentation, which can be Images and audio.

12.6 Summary

- Learned about features such as master slides to ensure uniformity and company identity of the presentations. Master Slides in PowerPoint enable you to keep a consistent outlook on your presentation in terms of design, layout, and formatting.
- Acquired the skills to include and utilise other media like audio and video in the presentation.
- Understood Advanced Animation Techniques & Acquired skills in animating to come up with lively and interactive displays.
- Delivering and printing presentations in PowerPoint involves more than just creating slides. From rehearsing timings to choosing the right print settings, we considered the possibilities of printing the presentations and putting them in different formats.
- Interactive presentations engage your audience and encourage participation, making your message more memorable and impactful. PowerPoint provides interactivity, including quizzes, navigation bars, and other elements that viewers can interact with.
- Discovered how to use high-level slide transitions and apply them in the process of making transitions in the slides.
- Discussed the options of using the tools and options available to a presenter to improve the presentation delivery process.
- Engaged in the use of collaborative technologies for teaming and collaborative writing of presentations.

12.7 Keywords

- **Animations: Animations** in MS PowerPoint are visual effects that apply movement or dynamic effects to text, objects, or even slides, creating interesting and dynamic presentations.

- **Multimedia Elements:** MS PowerPoint multimedia is a combination of media, which may include pictures, video clips, audio, and animation, used in presentations to make them attractive and informative.
- **Slide Show:** A slide show in MS PowerPoint is a series of slides arranged in a specific format so that they are shown one after another. Animations, multimedia objects, and transitions between the slides convey messages and information to the audience.

12.8 Self-Assessment Questions

1. Summarize your experience with master slides in MS PowerPoint. In what way do they manage to help in bringing out consistent presentation designs?
2. Describe a scenario where you used audio and video in your MS PowerPoint presentations and how you used them.
3. Explain some of the sophisticated animations that you have applied on MS PowerPoint in the event you were illustrating or trying to capture the attention of a group of people.
4. Can you explain how to deliver a presentation in MS PowerPoint and how you can manage it to make it more effective?

12.9 References / Reference Reading

- Lambert, James. Mastering Microsoft PowerPoint 2021: Tips and Tricks for Power Users. New York: Tech Publishers, 2022.
- Sinha, Ramesh. Advanced PowerPoint 2021: Techniques for Creating Professional Presentations. New Delhi: Prentice Hall India, 2023.
- Gupta, Neha. Interactive and Multimedia Presentations with PowerPoint. Bangalore: Wiley India, 2022.

Unit 13: Database Management Systems (DBMS)

Learning Outcomes:

- Students will be able to define key concepts of database management systems.
- Students will be able to explain different types of databases.
- Students will be able to compare various database models.
- Students will be able to apply SQL basics in database manipulation.
- Students will be able to design databases based on fundamental principles.

Structure:

13.1 Introduction to DBMS

A DBMS is a software system that provides the facilities for creating a database and modifying and retrieving data from the created database. It offers a structure for organising and accessing data competitively and efficiently that will enhance data security and integrity. Here's an overview of the key concepts and components of a DBMS

1. What is a DBMS?

Definition: A DBMS is an application that communicates with end-users, other applications, and the actual data storage to store and process data. It enables the storage, access and manipulation of information in a well-formatted manner.

Examples: Some of the widely used DBMS software are MySQL, PostgreSQL, Oracle, Microsoft SQL Server and SQLite.

2. Components of a DBMS

- Database Engine: The primary service used to get the data and use it according to the requirements.
- Database Schema: This describes how the data is organised logically, such as tables, fields, relationships, and constraints.
- Query Processor: Translates the query written in SQL (Structured Query Language) and performs it.
- Transaction Management: Responsible for guaranteeing that all database transactions are performed and are compliant with the ACID properties, which include atomicity, consistency, isolation, and durability.
- Storage Management: Responsible for the storage of data on the disk space.
- Database Administration Tools: These tools help comprehend the state of an unorganised database and assist with upkeep, tracking, and administration.

3. Functions of a DBMS

- Data Storage and Retrieval: Able to store huge amounts of data and retrieve the data upon request within a short time.
- Data Manipulation: It supports data insertion, update, deletion, and querying activities.
- Data Integrity: Maintains data validity and integrity through constraints as well as validation rules.
- Data Security: Secures the data and takes precautions to prevent its leakage or misuse by other people.

- Backup and Recovery: They enable data backup and recovery to minimise data loss.
- Multi-User Access: This allows two or more users to use the database at a given time without conflicting with each other while ensuring the integrity of the data stored in the database.

4. Advantages of Using a DBMS

- Data Redundancy Reduction: Reduces data redundancy since data collected is integrated into a single database.
- Data Sharing: Allows information to be accessed and transmitted by more than one user and application.
- Data Integrity and Accuracy: It guarantees the data and its uniformity through integrity constraints.
- Data Security: Safeguards data through the use of a username and password or any other security measures that may be adopted.
- Scalability and Flexibility: Flexible enough to accommodate increased data and changes in data demand or need.
- Efficient Data Access: Enhances the probability that data will be accessed as and when it is required by creating indexes and efficient query processing.

13.2 Types of Databases

Databases themselves can be of different types, and they all are created to work with certain types of data and applications. Knowledge of the various categories of databases aids in choosing the most appropriate one for the intended use. Here's an overview of the major types of databases:

1. Relational Databases

Definition: Data in relational databases is stored in tables known as relations, which are made up of rows and columns. They employ Structured Query Language (SQL) for querying and manipulating data.

- Examples: MySQL, PostgreSQL, Oracle, Microsoft SQL Server and so on.
- Use Cases: Applicable for activities that involve comprehensive queries and transactions, such as financial processing and ERP and CRM systems.

2. NoSQL Databases

Definition: SQL or NoSQL are a type of database that is capable of storing unstructured, semi-structured data with a free-form concept.

Types:

- Document Databases: Store data in JSON, BSON or XML documents such as MongoDB or CouchDB.
- Key-Value Stores: Save data as a key-value pair, such as Redis or Amazon DynamoDB.
- Column-Family Stores: Suggest storing data in columns rather than rows (e.g., Apache Cassandra, HBase).
- Graph Databases: Data should be stored in a graph model suitable for relations and connections (Neo4j, ArangoDB, etc.).
- Use Cases: These are most suitable for applications that need to develop and grow, analyse data in real-time, and work with a lot of unstructured data, including social networks, IoT, and big data.

3. Object-Oriented Databases

Definition: Object-oriented databases are databases in which data is stored as objects, mirroring the concept of objects used in object-oriented programming.

- Examples: ObjectDB, db4o, Versant.
- Use Cases: Applied in tasks that involve data modelling, for instance, computer-aided design (CAD), multimedia, and scientific uses.

- **Knowledge Check 1**

  Fill in the Blanks

  1. Understanding ___________ is fundamental for managing large sets of structured data. (database management)
  2. Types of databases include relational, ___________, and object-oriented databases. (NoSQL)
  3. Database ___________ serve as blueprints for organising and structuring data in DBMS. (schemas)
  4. SQL, or Structured Query Language, is used for ___________ and managing databases. (querying)

- **Outcome-Based Activity 1**

  In groups, research and prepare presentations on different types of databases (relational, NoSQL, hierarchical, object-oriented).

13.3 Database Models

Database models define the logical structure and organisation of data in a database management system (DBMS). They provide a framework for how data is stored, organised, and manipulated. Here are the main types of database models:

1. Hierarchical Model

Structure: Organises data in a tree-like structure with a single root and multiple levels of child records. Each child has only one parent.

Advantages:

- Simple to design and understand.
- Efficient for hierarchical data relationships, like organisational charts.

Disadvantages:

- Limited flexibility due to rigid structure.
- Difficult to reorganise and maintain.

Example Use Case: File systems and organisational structures.

2. Network Model

Structure: Similar to the hierarchical model, it allows multiple parent-child relationships, forming a graph-like structure.

Advantages:

- More flexible than the hierarchical model.
- Efficient for representing complex relationships.

Disadvantages:

- More complex to design and manage.
- Requires navigational queries.

Example Use Case: Telecommunications and transport networks.

3. Relational Model

Structure: Organises data into tables (relations) with rows and columns. Tables are linked using foreign keys.

Advantages:

- High flexibility and simplicity.
- Powerful query capabilities using SQL.
- Strong theoretical foundation ensuring data integrity and consistency.

Disadvantages:

- Performance can degrade with very large datasets or complex queries.
- Requires careful design to avoid redundancy and ensure normalisation.

- Example Use Case: Enterprise applications, financial systems, and e-commerce platforms.

4. Entity-Relationship Model (ER Model)

Structure: This uses entities (objects) and relationships to represent data. It is often used in the database design phase to create a conceptual blueprint.

Advantages:

- Intuitive and easy to understand.
- Provides a clear visualisation of data relationships.

Disadvantages:

- Not directly implementable in a DBMS.
- Requires translation to a relational model for implementation.
- Example Use Case: Database design and planning.

13.4 SQL Basics and Applications

Definition: SQL is a standardised language used to communicate with relational databases.

Functions: SQL is used to create and modify database structures, insert, update, and delete data, and retrieve data from the database.

1. SQL Syntax

- SQL statements are composed of clauses, expressions, and predicates written in a declarative form.
- SQL keywords are case-insensitive, but it’s common practice to write them in uppercase.

2. Data Definition Language (DDL)

DDL commands are used to define and modify the structure of the database and database objects.

CREATE: This operation creates a new table, database, index, or view.

```
SQL
CREATE TABLE Employees (
  EmployeeID INT PRIMARY KEY,
  FirstName VARCHAR(50),
  LastName VARCHAR(50),
  BirthDate DATE
);
```

- **ALTER:** Alters an existing database object, including a table.

SQL

```
ALTER TABLE Employees
ADD Email VARCHAR(100);
```

- **DROP:** Removes an object from a database, provided that it exists in the database.

SQL

```
DROP TABLE Employees;
```

3. Data Manipulation Language (DML)

DML commands are those commands that are used to modify the data stored in databases.

- **INSERT:** used to insert new records into the table.

SQL

```
INSERT INTO Employees (EmployeeID, FirstName, LastName, BirthDate)
VALUES (1, 'John', 'Doe', '1980-01-01');
```

- **SELECT:** Used to search for data in one or more tables.

SQL

```
SELECT FirstName, LastName
FROM Employees
WHERE BirthDate > '1980-01-01';
```

- **UPDATE:** Update information within a table, that is, change certain record values to their new ones.

SQL

```
UPDATE Employees
SET Email = 'john.doe@example.com'
WHERE EmployeeID = 1;
```

- **DELETE:** Avoids a table by deleting rows from it.

SQL

```
DELETE FROM Employees
WHERE EmployeeID = 1;
```

4. Data Control Language (DCL)

Security control commands are issued through DCL to regulate data access in the database.

- **GRANT**: Allows a user to do something specific.

```
SQL
GRANT SELECT, INSERT ON Employees TO ‘user_name’;
```

- **REVOKE**: Revoke the permission from a user.

```
SQL
REVOKE INSERT ON Emps FROM user_name;  SQL
  GRANT SELECT, INSERT ON Employees TO user_name
```

Applications

SQL is commonly used in numerous applications in different fields for the administration and manipulation of Relational Database Management Systems. Here are some common applications of SQL:

1. Web Development:

- Database Integration: SQL is employed for storing and accessing data in databases, and it runs many dynamic web applications. Django, Ruby on Rails, Laravel, and many other web frameworks use SQL for database operations.
- Content Management Systems (CMS): SQL databases store content on websites that let users build, modify, and manage websites and articles.

2. Business Intelligence (BI):

- Data Analysis: SQL can be used to pull data from big data sources, perform analysis, and make business decisions based on the results of the analysis.
- Data Warehousing: One possible application is data warehouse management, where SQL is used to build the structures and manage the collected data and its sources for analysis and reporting.

3. E-commerce:

- Inventory Management: SQL databases are used to keep records of products, control stock levels, and process orders and transactions in e-commerce systems.
- Customer Relationship Management (CRM): SQL databases are useful for storing customer data, purchase data, and interactions, especially in marketing and customer support.

4. Finance and Banking:

- Accounting: SQL databases handle financial transactions, keep records of expenditures, and prepare the accountants' financial statements.
- Online Banking: User accounts, transactions, and balance inquiries are managed in online banking systems by SQL databases.

5. Healthcare:

- Electronic Health Records (EHR): In health care systems, SQL databases help in storing patients' details, their records, and treatment details.
- Medical Research: SQL queries use medical data to analyse trends, patterns, or correlations required in medical research and clinical studies.

13.5 Database Design Principles

The principles of databases are very important when it comes to designing databases that can be very efficient and scalable, and that can be trusted. The principles help database architects and developers to organise data according to the needs of companies and their further usage, availability, and manageability. Here are the key principles of database design:

1. Understand the Requirements

Therefore, it is necessary to know the business requirements, such as the data that will be stored, how the data will be used, and the kinds of queries to be done ahead of the design. This way, the database will always be able to handle all the required operations and functions.

2. Normalisation

Normalisation is a process of structuring the data to reduce redundancy and increase the data quality. This involves partitioning a database into tables as well as establishing connections between those tables. The main normal forms (1NF, 2NF, 3NF, and BCNF) regulate this process and have particular rules that help to eliminate the redundancy of data and to distinguish dependencies.

3. Application of Primary and Foreign Keys

Primary keys define a unique column for each record in a table so that every entry in a table can be located and accessed individually. Foreign keys are used to set some constraints on tables to make sure that a value in one table matches a value in another table.

4. Efficient Indexing

Indexes are employed to enhance the rate at which data retrieval operations are conducted. They also serve the purpose of the index in a book such that the database can locate the data without searching the entire table. However, excessive indexes hinder the execution of data modification operations and, therefore, the need to balance it to achieve the best results.

5. Data Integrity

Data integrity means the quality aspect of data that revolves around the accuracy and comprehensiveness of data throughout its life cycle. This involves the use of constraints that include NOT NULL, UNIQUE, CHECK and FOREIGN KEY constraints to ensure business rules and data validation rules are observed at the database level.

- **Knowledge Check 2**

  Fill in the Blanks

  1. The principles of database design help to make the __________ convenient to store and access data. (efficient)
  2. In DBMS, data is organised into tables with rows and ___________. (columns)
  3. Depending on the requirements, different forms of databases are available, like ___________ databases for multimedia data. (multimedia)
  4. It is important to know database models such as the hierarchical model and the network model for ___________. (structures and relationships)
- **Outcome-Based Activity 2**

  Divide into groups of 2 and proceed through the SQL query practice on a sample database schema. Generate statements for selecting records, creating new records, modifying records and deleting undesirable records.

13.6 Summary

- Discussed the concepts and significance of DBMS in the aspect of efficient storage and organisation of information.
- I understood the concept of the database and various forms of, such as relational, NoSQL, hierarchical, and other forms, as well as the circumstances in which they can be applied.

- Database models define data's logical structure and organisation in a database management system (DBMS). They provide a framework for how data is stored, organised, and manipulated.
- Acquired knowledge of SQL to work with databases, manipulate data, and manage the objects of a database. SQL is a standardised language used to communicate with relational databases.
- Understood concepts of database design for creating sustainable, high-performing databases.
- SQL is commonly used in numerous applications in different fields for the administration and manipulation of Relational Database Management Systems. Here are some common applications of SQL: Web Development, Business Intelligence (BI), E-commerce, etc.
- Normalisation is a process of structuring the data to reduce redundancy and increase the data quality. This involves partitioning a database into tables as well as establishing connections between those tables. Studied the proportion of normalisation and its use in database design to minimise data repetition. I also Learned about the methods for indexing and optimisation of the databases and queries for faster access to the queries.
- As part of Security and Authorisations, we have understood the procedures of security and authorisation to prevent unauthorised access to databases and maintain confidentiality.

13.7 Keywords

- **DBMS:** DBMS is a system that provides interfaces to the end user, applications, and the database to retrieve/store/ process information.
- **NoSQL Databases:** NoSQL (Not Only Structured Query Language) is a type of database used to store data that is unstructured or semi-structured and has a nonfixed structure.
- **SQL:** SQL is an international standard used to interface with Relational Database Management Systems.

13.8 Self-Assessment Questions

1. What is the function of a database management system (DBMS) in the organisation and proper management of data? Also, how does it differ from traditional file-based systems?
2. What are the differences between Relational and NoSQL databases and their use cases?
3. Take a moment to describe other forms of database models with examples such as the hierarchical, network and relational types. When, where, and why might you choose one model over the other?
4. Explain why SQL is relevant in the communication process with databases. Would you mind giving an example of how one can practically apply SQL?
5. Explain the major concepts of proper database design and how such concepts facilitate the effectiveness and accuracy of a database.

13.9 References / Reference Reading

- Elmasri, Ramez, and Shamkant B. Navathe. *Fundamentals of Database Systems*. 7th ed., Pearson, 2016.
- Silberschatz, Abraham, Henry F. Korth, and S. Sudarshan. *Database System Concepts*. 7th ed., McGraw-Hill Education, 2019.
- Date, C. J. *An Introduction to Database Systems*. 8th ed., Pearson Education India, 2019.
- Ramakrishnan, Raghu, and Johannes Gehrke. *Database Management Systems*. 3rd ed., McGraw-Hill, 2003.
- Desai, Bipin C. *An Introduction to Database Systems*. Galgotia Publications, 2012.

Unit 14: Cybersecurity and Data Protection

Learning Outcomes:

- Students will be able to define fundamental cybersecurity concepts.
- Students will be able to explain common cyber threats and attacks.
- Students will be able to identify data protection techniques.
- Students will be able to apply best practices in cybersecurity.
- Students will be able to evaluate legal and ethical issues in cybersecurity.

Structure:

14.1 Introduction to Cybersecurity

Cybersecurity is the process of safeguarding computers, servers, mobile devices, electronic systems, networks, and data from malicious attacks, digital invasions, cyber threats, and other cyber dangers. It covers a broad area involving all the technologies, methods, and activities used to protect data and information from unauthorised access. Here's an overview of key concepts and components of cybersecurity:

# 1. Cyber Threats

Definition: Cyber threats, on the other hand, refer to possible risks and risk factors that can threaten the security of online resources.

Types of Threats:

- Malware: Computer programs with malicious intent to harm or infiltrate computer systems (for instance, viruses, ransomware, and spyware).
- Phishing: Phishing, vishing, smishing, etc.; cyber attacks that involve deception, such as impersonation, to get the targeted victim to divulge sensitive information like passwords or financial information.
- SQL - Distributed Denial of Service (DDoS): A technique used to inundate a network or a system with traffic, making it almost impossible to use for legitimate purposes.
- Insider Threats: Threats resulting from malicious insiders, users who exploit their authorised permissions to compromise an organisation’s assets or those with ill intentions towards the company.
- Advanced Persistent Threats (APTs): Advanced and strategic types of attacks that professional attackers normally conduct with the intention of gaining access to high-worth targets.

# 2. Cybersecurity Principles

- Confidentiality: ensuring that only authorised personnel or organisations can access specific information.
- Integrity: Protecting the integrity of a set of data and a system by inhibiting improper change by an unauthorised individual.
- Availability: Ensure that the systems, services, and data are always available and up and running as and when required.
- Authentication: Authentication as an aspect of access control is the process of confirming the identity of the users, devices, or entities seeking to gain access to the resources.

- Authorisation: Assigning the right level of permissions and access rights to the users who are authenticated and should perform specific tasks.
- Non-Repudiation: Eliminating the possibility of people denying certain actions or certain transactions that took place.

14.2 Common Cyber Threats and Attacks

Basic Forms of Cyber Threats and Assaults

These threats are real and present a very grave danger to anyone who uses computers and the internet or is in any way connected to the digital world. These threats take advantage of the weak points in the computers, networks, and people to steal, damage, or gain control of information and negatively affect the business financially or otherwise. Here are some of the most prevalent cyber threats:

1. Malware

Definition: Malware is a shortened term for malicious software, which is a program designed to cause harm and exploit computers, networks, and devices.

Types:

- Viruses: Those that copy themselves and spread to other files or other systems.
- Trojans: Applications that look otherwise are actually designed to perform some nasty activities when run.
- Ransomware: A type of software that hides or secures files or the operating system and then requests money for the files to be unlocked or the system to be unfrozen.

2. Phishing

Definition: Phishing is a form of social engineering in which people are sent messages or directed to websites that mimic the look and feel of legitimate organisations. The websites then entice people to reveal personal information such as passwords, credit card numbers, or other personal details.

Types:

- Email Phishing: Phishing emails that require the receiver to send cash, credit card numbers, passwords or other personal information to fake businesses or persons.
- Pharming: Phishing websites or Domain Name Systems (DNS) that take users to fake websites to enter their login details or credit card details.
- Impact: phishing can result in identity theft, financial loss, unauthorised access to other accounts, and damage to reputation.

3. Other attacks include Distributed Denial of Service (DDoS).

Definition: They overwhelm a target system, network, or service with a large amount of traffic or requests, making it unavailable to genuine users.

Types:

- Volumetric Attacks: Overloads the target with a huge traffic load, putting pressure on the network resources.
- Protocol Attacks: Uses vulnerabilities in protocols like the TCP/IP to interfere with communication between the devices.
- Application Layer Attacks: Floods specific applications or services with malicious requests/traffic to overwhelm them.
- Impact: DDoS attacks can interrupt the normal use of various online services, leading to unavailability and slowness and, in the long run, losses for firms.

- **Cyber Attacks**

Cyber crimes refer to unscrupulous acts aimed at creating weaknesses in computers, computer networks, and the conduct of people. These attacks are executed to obtain confidential information to complicate the company's functioning or to seek economic or image losses. Here are some of the most prevalent types of cyber attacks:

1. Malware Attacks

Definition: Malware attacks are instances where an attacker uses a virus or other malicious code to disrupt the functionality of computers, networks, or other devices.

Types:

- Viruses: A computer virus that must be run by the user and copies itself into other files or systems.
- Worms: A type of virus that can initiate itself on networks without the user's approval, and it infects computers by targeting software flaws.
- Trojans: Viruses are programs that are embedded in other programs and are capable of running on their own with the intention of causing harm.
- Ransomware: A virus that locks an operating system or encrypts files and then asks for an amount to be paid to unlock or decrypt the item.
- Impact: The risks of malware attacks involve loss of data, time, money as well as reputation.

2. Phishing Attacks

Definition: Phishing scams employ fake emails, messages, or websites to obtain personal information from a user, such as a password, credit card number, or identity number.

Types:

- Email Phishing: Phishing emails is an act in which fake emails are created in the name of an organisation or a competent personality in an effort to deceive the recipients.
- Spear Phishing: A phishing attack is targeted at certain persons or companies and launched using information that can convince the targeted persons.
- Pharming: Fraud sites or fake DNS attacks lead users to phoney sites where they can enter their credentials or monetary details.
- Impact: The losses arising from phishing include identity theft, financial losses, unauthorised access to accounts, and damaging corporate image.

3. Distributed Denial Of Service (DDoS) Attacks

Definition: During a DDoS attack, the attacker sends a large number of requests or traffic to the target system, network or service, preventing valid users from getting through.

Types:

- Volumetric Attacks: Flood targets with high traffic volumes that put a lot of pressure on the program's resources.
- Protocol Attacks: Takes advantage of vulnerabilities in the TCP/IP protocol to interfere with the transmission of information between connected systems.
- Application Layer Attacks: These attacks target certain applications or services, flooding them with requests and creating a denial of service.
- Impact: These attacks can interrupt normal operations of online services, bring websites and/or services down, slow down performance, and cost money.

- **Knowledge Check 1**

  Fill in the Blanks

  1. Speaking about cybersecurity basics, it is necessary to mention that it deals with the protection of ___________ and other pertinent data. (networks)
  2. Other threats and cyber attacks range from ___________ and phishing. (malware)

3. Methods of data protection include encryption and ____________ to protect the data from tampering. (integrity checks)
4. Some of the strategies that are considered to be effective in cybersecurity are ____________ and training of employees. (software update)

- **Outcome-Based Activity 1**

  List down frequent types of cyber threats and attacks like viruses, phishing, and ransomware. Explain how they work, what effects they have, and how they can be prevented.

14.3 Data Protection Techniques

1. Encryption

Definition: Encryption, on the other hand, is the conversion of plain text data into cypher text using cryptographic algorithms and keys to make the data indecipherable by those who are not supposed to access it.

Types:

- Symmetric Encryption: Works with the same key and is good for encrypting data that will be stored in the database or other storage means.
- Asymmetric Encryption Uses both a public and a private key to encrypt and decrypt messages, which is useful in secure communication and key exchange.
- Applications: Protection of data in the form of files, databases, communication, and media storage and preventing unauthorised access.

2. Access Controls

Definition: Controls limit and monitor users' access rights to data and resources in accordance with their identities, functions, and authorities.

Techniques:

- Authentication: Confirming the legitimacy of users using credentials such as passwords, fingerprints, or token credentials.
- Authorisation: Providing specific access to the application for credentialed users based on their roles in the organisations.
- Multi-Factor Authentication (MFA): This method forces users to provide multiple individual checkpoints to gain access to certain data or applications.

3. Data Backup and Recovery

Definition: Data backup and recovery refer to the processes of creating duplicates of vital data and their storage for quick accessibility and protection from loss or damage.

Techniques:

- Regular Backups: Automated or periodic copies of data to secondary storage media or cloud storage.
- Incremental and Differential Backups: Saving updated data or new data only for the backup since a backup can take time and space, especially if it was done a long time ago.
- Offsite and Cloud Backup: Having multiple copies at sites in different geographical regions or on the cloud for business recovery or continuity.
- Applications: Preserving against data loss resulting from hardware failure, floods, viruses or hackers, or deletion by an employee.

14.4 Best Practices of CyberSecurity

Cyber security guidelines are very crucial in safeguarding systems, networks, and data against cyber threats and attacks. These practices incorporate a variety of measures, including preventive measures, security controls, and tactics that are employed to minimise risks and develop a good security system. Here are some key best practices in cybersecurity: Here are some key best practices in cybersecurity:

1. User Education and Awareness

- Training Programs: It is also important to conduct periodic training sessions and create awareness among the employees regarding the common threats, phishing attacks, and security measures.
- Policy Enforcement: Security policies, guidelines, and acceptable use of IT systems should be set and complied with to reduce security incidents.

2. Patch Management

- Regular Updates: The patch management best practice should be followed to ensure that all operating systems, software, and firmware are updated with the latest patches and fixes to known vulnerabilities.
- Vulnerability Scanning: Perform vulnerability assessments and scans to know which vulnerabilities are suitable for addressing and fixing.

3. Access Controls

- Least Privilege: Follow the principle of least privilege, which limits user privileges to what is needed to complete the user's task by barring them from having excessive permissions that may lead to vulnerabilities if compromised.
- Strong Authentication: Use MFA methods to ensure that the user and the device are real and not fake by stepping up beyond the use of passwords.

4. Network Security

- Firewalls: Secure the system by installing firewalls and intrusion detection/prevention systems (IDS/IPS) to filter traffic and prevent unauthorised traffic access in the networks.
- Segmentation: Segment networks into different zones or subnets according to the level of security needed to prevent breaches and minimize attacker movement.

5. Data Encryption

- Data Protection: Proper data encryption methods can be employed to protect data both at rest and in motion so that even with a breach, the information will remain safe.
- Secure Communication: Implement the use of TLS/SSL to ensure the transfer of data over the network, especially when dealing with essential information such as money transactions or personal information.

14.5 Legal and Ethical Issues in Cybersecurity

Cybersecurity legal and ethical concerns are important in establishing the aspects of risks that surround information security and what roles are played. Hence, it is crucial to fix these problems for organisations, governments, and people as cyber threats increase in number and complexity. Here are some of the key legal and ethical considerations in cybersecurity: Here are some of the key legal and ethical considerations in cybersecurity:

- **Legal Issues in Cybersecurity**

1. Data Protection & Privacy Laws

- General Data Protection Regulation (GDPR): The GDPR regulates data protection and privacy in the EU and has rigorous provisions for handling PD.
- California Consumer Privacy Act (CCPA): Like the GDPR, the CCPA offers several rights to California residents concerning their personal information and data: the right to know and the right to delete.

2. Compliance Requirements

- Health Insurance Portability and Accountability Act (HIPAA): HIPAA sets requirements for safeguarding health-related data in the United States of America.
- Payment Card Industry Data Security Standard (PCI DSS): PCI DSS is a security benchmark for companies that handle credit card information.

3. Intellectual Property Rights

- IP in the digital world translates to the safety of business inventions, written and artistic creations, and innovations against cyber theft and subsequent use without the owner's permission.

4. Cybercrime Legislation

- Different legislations relate to cybercrimes, including hacking, identity theft, and malware spreading. One such legislation is the Computer Fraud and Abuse Act (CFAA) in the United States of America.

- **Ethical Issues in Cybersecurity**

1. Privacy vs. Security

- This work identifies one of the major ethical issues: how to foster security without infringing on people's right to privacy. Protection precautions, including surveillance and data gathering, are involved in privacy violations.

2. Ethical Hacking

- White hat hackers are people who perform hacking for the purpose of determining the weaknesses in the security systems and helping in their rectification. However, they should work legally and should have the necessary permissions to do so.

3. Responsibility to Disclose Vulnerabilities

- There are moral questions as to when and how one should report the identified vulnerabilities. If reported within a Short time, it can expose systems to attacks; however, delayed disclosure can put users at risk.

4. Cybersecurity and Discrimination

- The measures that should be taken to ensure cybersecurity should not be discriminative in any way. For example, surveillance cannot be in a way that discriminates against certain people based on their race, ethnicity, etc.

This aims at establishing a balance of legal and ethical concerns.

- Developing Comprehensive Policies: Designing policies that conform to the laws and legal regulations and sticking to the right ethical principles.
- Promoting a Culture of Ethics: Promoting or enhancing ethical conduct and/or ethical management in organisations.

- Continuous Training: Informing cybersecurity specialists about changes in the laws and the ethical aspects.
- Stakeholder Engagement: Engaging a broad number of actors in the communication concerning cybersecurity measures and strategies.

When legal aspects are dealt with alongside ethical ones, organisations can establish trustful relationships, secure their data, and be part of creating a safer environment online.

- **Knowledge Check 2**

  Fill in the Blanks

  1. The legal and ethical issues in cybersecurity are defined by ___________ and privacy. (regulations)
  2. Multi-factor authentication enhances _____________ to an organisation’s computer networks and information. (security)
  3. Data ___________ and backups are necessary for the process of countering cyber threats. (encryption)
  4. Cybersecurity policies and procedures’ main goal is to ___________ and manage potential risks and weaknesses. (mitigate)

- **Outcome-Based Activity 2**

  Research cybersecurity laws and regulations such as GDPR, HIPAA, and PCI DSS. Divide the class into teams and assign each team a specific law or regulation to debate its importance and effectiveness in protecting individuals' privacy and organisations' data.

14.6 Summary

- Acquired knowledge about cybersecurity ideas and threats and recognised the risks associated with them. Cybersecurity is the process of safeguarding computers, servers, mobile devices, electronic systems, networks, and data from malicious attacks, digital invasions, cyber threats, and other cyber dangers.
- Cyber threats, on the other hand, refer to possible risks and risk factors that can threaten the security of online resources. Typical cyber threats include malware, phishing, ransomware, etc., and their effects.

- Phishing is a form of social engineering where by people are sent messages or directed to websites that mimic the look and feel of legitimate organisations and then entice people to reveal personal information such as passwords, credit card numbers, or other personal details.
- Encryption, on the other hand, is the conversion of plain text data into cypher text using cryptographic algorithms and keys to make the data indecipherable by those who are not supposed to access it.
- Cyber security guidelines are crucial in safeguarding systems, networks, and data against cyber threats and attacks. These practices incorporate a variety of measures, including preventive and security controls.
- Cybersecurity legal and ethical concerns are important in establishing the risks surrounding information security and the roles played. Hence, it is crucial to fix these problems for organisations, governments, and people as cyber threats increase in number and complexity.

14.7 Keywords

- **Cybersecurity:** It is defined as the protection of computer systems, networks and their associated data against unauthorised access, attack and malfunction.
- **Malware:** This is a term for malicious software designed to harm or exploit computers, networks, and devices.
- **Data Loss Prevention:** Data Loss Prevention (DLP) solutions involve products that identify and control the transfer of confidential information over the network, on the endpoints, and in storage devices.

14.8 Self-Assessment Questions

1. What measures can persons and companies take to avoid or prevent usual threats and cyber-attacks?
2. List some of the measures that one or any organisation can take in order to improve the security of their systems.
3. some legal and ethical issues should be taken into consideration when deploying security measures for cyberspace.
4. In what ways do cybersecurity awareness and training programs assist in decreasing security threats and cultivating a security mindset in an organisation?

14.9 References / Reference Reading

- Jangra, Sameer, et al. *Cyber Security and Ethical Hacking*. BPB Publications, 2021.
- Kumar, R.K. Shiroor, and Sameer Hiremath. *Cyber Security & Cyber Laws*. Dreamtech Press, 2022.
- Singh, Brij B., et al. *Cybersecurity and Cyberwar: What Everyone Needs to Know*. Oxford University Press, 2022.
- Subramanian, Ramesh. *Information Security: Principles and Practice*. Pearson Education, 2021.

Unit 15: Cloud Computing

Learning Outcomes:

- Students will be able to define cloud computing.
- Students will be able to explain types of cloud services (IaaS, PaaS, SaaS).
- Students will be able to identify the benefits and challenges of cloud computing.

- Students will be able to apply cloud computing applications.
- Students will be able to assess cloud security and compliance issues.

Structure:

15.1 Introduction to Cloud Computing

Cloud computing is an innovative model in the sphere of information technologies that has evolved the use of computing resources by companies and people. In its essence, cloud computing means that a large number of services, including servers, storage, databases, networks, applications, analytics, and even intelligence, are provided through the Internet, or ‘the cloud’.

1. # What Is Cloud Computing?

Cloud computing allows users to freely access and use data and applications through servers that are not in their local systems or local organisation servers. This leads to better flexibility, scalability, and utilisation of IT resources to support organisational goals and objectives. Cloud services are mainly provided by third-party vendors, including Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform, who are responsible for the backend support.

# Key Characteristics of Cloud Computing

1. On-Demand Self-Service: The computing resources can be provisioned by the users in an automated manner, so the service provider does not need to get involved.
2. Broad Network Access: Cloud services can be accessed from almost any Internet-enabled computing device, including laptops, mobile phones, tablets and more.
3. Resource Pooling: Cloud providers employ the multi-tenant model, where they share the customer’s application, data, and storage on the same physical infrastructure, though on a virtual basis based on usage.
4. Rapid Elasticity: The use of cloud services enables applications to increase and reduce their capabilities to meet the demands of customers, making them seem endless.
5. Measured Service: Cloud systems self-manage and self-optimise resource consumption using a metering function that permits visibility and billing.

Deployment Models in Cloud Computing

1. Public Cloud: Services are accessed and utilised over the Internet, and different organisations use them. For instance, there is AWS and Microsoft Azure.
2. Private Cloud: This is because services are run on a private network, which is preferred in business organisations that have unique regulatory or security demands.
3. Hybrid Cloud: Integrates both the public and the private cloud, where the data and the applications can be transferred from one to the other. This gives more flexibility and utilisation of the existing structures, security, and compliance.

4. Community Cloud: This is consumed by several organisations and enables a specific group with similar needs, such as compliance needs. The organisations or a third party can administer it.

The following are the benefits of cloud computing;

1. Cost Efficiency: Pay-as-you-go financing eliminates the need to purchase capital equipment such as computers, other hardware, and software applications. This means you only pay for what you receive.
2. Scalability: When needs increase or decrease, the system can be quickly adjusted without incurring major costs in physical structures.

15.2 Types of Cloud Services (IaaS, PaaS, SaaS)

Cloud computing is typically divided into three main service models. For this application, there are three main categories of cloud computing services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). In this way, all the models present various degrees of control, flexibility, and management.

Infrastructure as a Service (IaaS)

aaS offers computing infrastructure through the internet in a virtual manner. This model enables users to hire infrastructure items like servers, storage, and networks in a usage-based model. It is suitable for organisations that require frequent expansions to their systems since they can easily provide this without having to invest in physical equipment.

Examples:

- Amazon Web Services (AWS) EC2: Presents the ability to obtain varying amounts of computing resources in the cloud.
- Google Compute Engine: Offers virtual machines hosted in Google's data centres.
- Microsoft Azure Virtual Machines: Makes it possible for users to run and control virtual machines in the cloud.

Benefits:

- Scalability: It is easy to expand or reduce the home business depending on the flow of customers in the market.
- Cost-Efficiency: Do not pay for the utilities that have not been consumed.

- Flexibility: Select the OS and application software as well as the hardware requirements and set-up.

Platform as a Service

PaaS provides hardware and software solutions through the Internet, mainly for application development. This model offers a foundation that enables developers to develop, test, and deploy applications without having to deal with physical hardware. PaaS is ideal for developers in that they do not have to worry about the physical aspects of building development, only the applications.

Examples:

- Microsoft Azure App Services: It provides the opportunity to create and deploy web applications.
- Google App Engine: Enables developers to write and run their code within Google's environment.
- Heroku: A multi-linguage multi-framework cloud platform.

Benefits:

- Development Speed: Create and deploy applications fast.
- Reduced Complexity: Emphasis on construction for development without having to consider the frameworks.
- Integration: They can easily be integrated with different other databases as well as development tools.

### ># SaaS – Software as a Service

SaaS involves licensing software applications to customers on a subscription basis over the Internet. This model enables users to use the software from any device with internet connectivity. The service provider is involved with the maintenance of the servers, the software, and security. SaaS can be considered most suitable for businesses that seek to implement their software solutions rapidly.

Examples:

- Google Workspace: Some of the G-suites are Gmail, Google Docs, and Google Drive.
- Salesforce: A case for a customer relationship management (CRM) system.
- Microsoft Office 365: Assists in gaining access to Microsoft Office Suite: Word, Excel, and PowerPoint through the internet.

Benefits:

- Accessibility: The user can access the applications by logging into the system from any location with an internet connection.
- Automatic Updates: Never use a software version that does not have automatic updates installed.
- Cost-Effective: Lower the initial investment amount possible through subscription-based pricing.

- **Knowledge Check 1**

  Fill in the Blanks

  1. This module on introduction to cloud computing defines the provisioning of on-demand __________. (computing resources.)
  2. Categories of cloud services are infrastructure as a service (IaaS), platform as a service (PaaS), and __________. (software as a service (SaaS))
  3. Facets of cloud computing include ___________ scalability and cost advantages. (elastic)
  4. Cloud computing finds its use in ____________ storage to software development. (data)

- **Outcome-Based Activity 1**

  Explain the Types of Cloud Services and their advantages in the classroom.

15.3 Benefits and Challenges of Cloud Computing

Cloud computing has many benefits; therefore, it is possible to conclude that it is beneficial for every business and its employees. Yet, it also has some peculiarities that should be taken into account as they may pose some problems.

# Benefits of Cloud Computing

1. Cost Efficiency: Lower Initial Costs: Companies can avoid hardware and software purchase costs, the establishment of internal data centres, and their operating expenses.
2. Scalability: Scalability means Increasing or decreasing resource use depending on the need in order to use the resources efficiently.
3. Flexibility—Remote Access: Employees can work from anywhere with an internet connection, which encourages telecommuting and information sharing.

4. Disaster Recovery—Robust Backup Solutions: Backup and disaster recovery are important for preserving business process integrity in the case of data loss or system failures.
5. Automatic Updates - Latest Technology: Cloud service providers often upgrade the systems with current technologies and security firms and patches.

Challenges Of Cloud Computing

1. Security and Privacy:

- Data Breaches: Off-premises storage of data exposes it to potential attacks from hackers and any other unauthorised personnel.
- Compliance: Data compliance, which relates to compliance with regulatory standards and standards that apply to a specific industry.

2. Downtime and Reliability:

- Service Outages: Cloud services are generally susceptible to outages and downtime, which are unbeneficial for businesses.
- Dependency on the Internet: A stable Internet connection is critical for service delivery, especially for cloud services.

3. Cost Management:

- Unexpected Costs: Though cloud computing reduces capital expenses, the variable cost can be very high if not controlled.
- Billing Complexity: Cloud service billing can also be complicated and overwhelming when there are various services and providers to deal with.

4. Vendor Lock-In:

- Limited Portability: Cloud migration can be difficult because data and applications must be compatible and interoperable between cloud service providers.
- Dependence on a Single Provider: The problem is that using one provider can be problematic if the provider's terms have been changed or if there are service interruptions.

5. Performance and Latency:

- Latency Issues: Where real-time processing is needed, there is likely to be a delay, especially where the data centres are far from the user.
- Bandwidth Limitations: Restricted bandwidth is a disadvantage because it influences cloud applications' functioning, especially when one is in an area with a slow internet connection.

15.4 Applications of Cloud Computing

Cloud computing has been used in several industries and has brought new ideas and improvements to flexible, scalable, and accessible solutions. Here are some of the key applications of cloud computing across different sectors:

# 1. Data Storage and Backup

- Scalable Storage Solutions: Some of the cloud storage services include Amazon S3, Google Cloud Storage, and Microsoft Azure Blob Storage for data storage with a flexible and dependable design.
- Automated Backups: Automatic backup systems should be implemented so as to ensure that the data is recoverable after such mishaps as hardware breakdowns or other calamities.

# 2. Disaster Recovery

- Cost-Effective Recovery Plans: Disaster recovery solutions that are deployed in the cloud are relatively cheaper and more efficient ways of performing data recovery and continuity.
- Reduced Downtime: AWS Disaster Recovery and Azure Site Recovery are some of the methods that make it possible to reduce downtime since the data and application are replicated in real-time.

# 3. Software Development and Testing

- Development Environments: AWS Elastic Beanstalk, Google App Engine, and Azure DevOps are examples of platforms that allow and enhance the development environments for the software development life cycle.
- Collaboration Tools: These allow users to work in real time and share code and projects to improve teamwork and productivity.

# 4. Big Data Analytics

- Data Processing: Hadoop, Hive, AWS Redshift, Google BigQuery, and Azure Synapse Analytics are some of the most effective platforms and tools for analysing big data.
- Real-Time Analytics: Cloud services contain all the necessities for real-time data analysis, which is useful when daily business decisions are made.

# 5. Automated Intelligence and Machine Learning

- AI and ML Services: AWS offers services such as AWS SageMaker, Google offers Google AI Platform, and Microsoft offers Azure Machine Learning for the development, training, and use of machine learning models.
- Natural Language Processing (NLP): Tools like Google Cloud Natural Language API and AWS Comprehend are available services that offer NLP abilities such as sentiment analysis, text classification and others.

15.5 Cloud Security and Compliance

The need for security and compliance in the cloud environment has become essential this year as more organisations adopt cloud computing services. Cloud security aims to safeguard data, applications, and services against threats and malicious attacks, while compliance relates to the legal requirements that need to be met in a certain business.

# Cloud Security

1. Data Protection:

- Encryption: This also means that data has to be encrypted during transfer and stored in different mediums. This will guarantee that in the event that data is captured or accessed by an unauthorised person, it will still be in an unreadable format.
- Access Controls: Strengthening measures like MFA and RBAC safeguard data from being accessed by unauthorised people.

2. Identity and Access Management (IAM):

- User Authentication: Security mechanisms of authentication allow only permitted users to access cloud resources.
- User Authorisation: Authorisation policies specify which assets a subject can use and what operations can be done on them.

3. Network Security:

- Firewalls: Cloud firewalls inspect and filter traffic through the cloud network, depending on the security policies that have been implemented.
- Virtual Private Networks (VPNs): VPNs establish a secure communication link between cloud resources and on-premise platforms.

4. Security Monitoring and Threat Detection:

- Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS): These systems identify security threats and take action.
- Security Information and Event Management (SIEM): SIEM solutions collect and analyze security-related information to identify threats and respond to them.

5. Data Loss Prevention (DLP):

- Backup and Recovery: It also focuses on data availability and data integrity through data backups and disaster recovery processes.
- Data Redundancy: Distributing data to more than one place in order to avoid data loss quality due to the failure of one or more hardware units.

## Compliance in Cloud Computing

1. Regulatory Compliance:

- General Data Protection Regulation (GDPR): This regulation preserves EU citizens' personal data and impacts any organisation that processes or stores such data.
- The Health Insurance Portability and Accountability Act (HIPAA) preserves the privacy of patients' health information in the healthcare industry.
- Payment Card Industry Data Security Standard (PCI DSS): This standard develops guidelines for organisations that offer credit card services.

2. Industry Standards and Frameworks:

- ISO/IEC 27001 provides the guidelines for implementing, maintaining, and continually enhancing an organisation's information security management system (ISMS).
- SOC 2: Describes how to implement customer data management based on five "trust service principles" such as security, availability, processing integrity, confidentiality, and privacy.

3. Compliance Strategies:

- Vendor Assessment: Overseeing the matters of compliance of the CSPs with the legal requirements and the best practices of the industry.
- Data Governance: >Set up the guidelines and protocols regarding data security, privacy, and integrity.
- Continuous Monitoring: Continuing to scan cloud environments to make sure regulatory compliance is maintained on going.

4. Data Residency and Sovereignty:

- Data Residency: Removes data when it is located in a certain geographical location as the law provides for.
- Data Sovereignty: Responsible for the local laws and regulations of the country where the data is stored.

- **Knowledge Check 2**

  Fill in the Blanks

  1. Cloud security and compliance deal with the ___________ of data and conformity to regulations. (protection)
  2. Cloud migration involves the moving of ___________ and applications to a cloud environment. (data)
  3. Hybrid cloud is designed to provide flexibility by integrating __________ with the private cloud. (public cloud)
  4. the cloud providers provide SLAs with the aim of guaranteeing performance and _______. (availability)

- **Outcome-Based Activity 2**

  Form a group and then debate on the topic 'Advantages and Disadvantages of Cloud Computing'.

15.6 Summary

- Provided an understanding of cloud computing, which is defined as a computing paradigm, and discussed its features and cloud deployment models. Understood IaaS, PaaS, and SaaS concepts with hosting, service, and software as types of Cloud Services
- Cloud Computing has benefits such as scalability, flexibility, and cost advantages. Its challenges include security and compliance issues and data privacy.
- Cloud Deployment Models are public, private, hybrid and multi-cloud deployment
- Cloud computing has found its use in several industries and has brought in new ideas and improvement to the flexible, scalable and accessible solution. A few of the applications are Automated Intelligence and Machine Learning, Big Data Analytics, etc

- Cloud security aims to safeguard data, applications, and services against threats and malicious attacks, while compliance relates to the legal requirements that need to be met in a certain business.
- Cloud Compliance: Identified key compliance concerns concerning cloud computing, such as GDPR, HIPAA, etc.
- Gained knowledge on how to migrate applications and data to the cloud and the recommended good practices.
- Discussed some new and prospective solutions and developments in cloud computing, such as serverless computing, edge computing, etc.

15.7 Keywords

- **Cloud computing:** This presents a model whereby computer utility is provided by storing data and applications online instead of the local computer or on-premises server.
- **AWS:** Amazon Web Services, Inc. (AWS) is an internet subsidiary of Amazon.com which offers hosted, affordable web services and APIs to businesses and individuals on a demand basis.
- **Customer Relationship Management:** CRM (Customer Relationship Management) is the approach to managing the organisation's relationships with customers. CRM is a business strategy that seeks to understand and manage all points of customer contact throughout the customer's life cycle.

15.8 Self-Assessment Questions

1. Define cloud computing and its importance in the context of the contemporary information technology environment.
2. Explain the differences and similarities of the three models of cloud services, namely IaaS, PaaS, and SaaS and give examples of cloud applications in each of the models.
3. Evaluate the advantages and disadvantages of cloud computing in the business world.
4. Include cases of organisations using cloud computing in real life in various sectors (e.g., healthcare, finance, e-commerce).

5. How do cloud security and compliance measures ensure data confidentiality, integrity, and availability in the cloud environment?

15.9 References / Reference Reading

- Buyya, Rajkumar, Christian Vecchiola, and S. Thamarai Selvi. Mastering Cloud Computing: Foundations and Applications Programming. Morgan Kaufmann, 2013.
- Bansal, Vipin, and Manisha Bharti. Cloud Computing: Concepts and Technology. BPB Publications, 2020.
- Srinivasan, S., and R. J. D'Souza. Cloud Computing: A Practical Approach for Learning and Implementation. Pearson Education India, 2014.
- Pallis, George, and Athanasios V. Vasilakos. Cloud Computing: Theory and Practice. Morgan Kaufmann, 2014.
- Thamarai Selvi, S., and G. Sudha Sadasivam. Cloud Computing: Principles and Paradigms. Wiley India Pvt. Ltd., 2011.

Unit 16: Emerging Technologies in Computing

Learning Outcomes:

- Students will be able to define AI and ML concepts.

- Students will be able to explain the role of IoT in modern computing.
- Students will be able to describe blockchain technology.
- Students will be able to identify future trends in computing.
- Students will be able to evaluate the potential impact of quantum computing.

Structure:

16.1 Artificial Intelligence (AI) and Machine Learning (ML)

Understanding Artificial Intelligence (AI)

AI is the operation of machines in a way that is comparable to human intelligence in that it learns, reasons, and makes decisions, among other things. This umbrella term covers such things as learning ability, the ability to reason, solve problems, perceive, and understand language. AI systems can be classified into two types: There is a narrow AI intended for a particular task, for example, face recognition or search for information on the Internet, and general AI, which can solve any problem that is within a person's capabilities.

The history of AI dates back to the mid-twentieth century when John McCarthy created the term. In its early stages, AI-based research was based on symbolic methods and problem-solving. The field has developed over the years as it has adopted new computing capabilities, increased data availability, and better algorithms.

Exploring Machine Learning (ML)

There is Machine Learning (ML), a branch of AI that focuses on a computer's ability to learn on its own and become wiser through practice. In contrast to the typical imperative programming, where the computer is told step by step how to solve a problem, in ML, a model is trained on a dataset, and when new data come in, the model makes a decision or a prediction without being programmed for that immediately.

ML can be broadly categorised into three types:

1. Supervised Learning: In this approach, the training data is labelled, which means that each training example has an output label. The model learns to forecast the output from the input data. Some of the applications include classification and regression problems.
2. Unsupervised Learning: Here, the model is fed information, and it does not know what to do with it, which means it does not have any specific duties or responsibilities. The objective of this model is to understand the patterns and structure of such data. Some of the most used methods are clustering and dimensionality reduction.
3. Reinforcement Learning: This type of learning takes place when an agent is trained to make a sequence of decisions by being given a reward or punished for an action. Over time, the agent learns to acquire the biggest sum of rewards.

AI and Machine Learning Key concepts

1. Algorithms

It is a series of procedures or instructions that are provided to an AI or ML model to enable it to find insights from data. It is the foundation of any AI or ML solution and helps systems accept input, analyse it, and generate results.

2. Data

Information is the most important aspect of AI and ML. It is made up of the data that is actually used to feed algorithms and build models. Structured data includes databases, while unstructured data includes text and images, among others. The quantity and quality of data affect the performance of AI and ML.

3. Training and Testing

Training in the scope of ML implies utilising a dataset to introduce the ability to make predictions or decisions in a model. This process involves evolving the parameters of the model to suit the results obtained from it. Testing is the process of assessing the accuracy of the model on a new data set to check the model's performance.

4. Features

Feature is the independent variable that is plugged into the model to help conclude. Features in a dataset are the columns, and each line refers to a record. They named the procedure of choosing the subset of features that should be used in the model as a feature selection.

5. Models

A model is one of the outcomes of training in the case of ML. It is a mathematical model derived from the data that can be used to forecast or make a decision. The models can range from simple linear models to the more complex neural networks model.

6. Supervised Learning

Supervised learning is one of the ML techniques where the model is trained, and each training example is associated with an output label. The model builds a relationship between the input data and the output it expects to obtain. For instance, classification is the process of grouping data, while regression is estimating numerical values.

7. Unsupervised Learning

As for the second one, unsupervised learning is the process of training models on the data for which no responses are available. The aim is to get insights into the data analysis and to look for relationships within the data set. Some of the most used techniques of unsupervised learning are clustering, which involves grouping similar data, and dimensionality reduction, which consists of decreasing the size of the feature vector.

8. Reinforcement Learning

Reinforcement learning is a branch of ML in which an agent learns how to act in an environment to maximise rewards or minimise penalties. The agent's goal is to accumulate rewards over the longest period possible, and in this process, the agent has to make corrections.

AI and ML are used in algorithms, as discussed in the following sections.

1. Linear Regression:

- Example: Regression analysis helps identify how some features of a house, such as the house's size, the number of bedrooms, or its location, can be used to estimate the price of the house.
- Application: Market research and prediction of the stock market.

2. Decision Trees:

- Example: Determining whether an email is spam or not based on the presence of certain attributes, such as certain keywords.
- Application: C. Email filtering systems.

3. Support Vector Machines (SVM):

- Example: Recognising handwritten digits by clustering in a high-dimensional layer with different classes.
- Application: Optical Character Recognition, sometimes referred to as optical machine recognition.

16.2 Internet of Things (IoT)

What is IoT?

The IoT is the connection of things of a physical nature, such as vehicles, buildings, and other objects, with sensors, software, and network connectivity through which these objects can gather and share information. The main purpose of IoT is to connect the physical world and the cyber world, where devices can share data, interpret, and respond to it.

Key Components of IoT

1. Sensors and Actuators: Sensors gather information from the surroundings and relay it where aspects like temperature, humidity, light, movement, and others are

involved. Executors take action on the data, such as adjusting a valve or switching on a light.

2. Connectivity: Devices also require a way to broadcast to other devices and a central system. Different communication interfaces and technologies, such as Wi-Fi, Bluetooth, Zigbee, cellular networks, etc., can achieve this.
3. Data Processing: Collected data is analysed either at the edge, which is closer to the source, or in the cloud. This processing can range from data sorting to data analysis.

How IoT Works

1. Data Collection: Sensors, as well as devices, acquire data from their surroundings.
2. Data Transmission: This data is sent over a network to a central system and may be passed through gateways when the two systems employ different types of communication.
3. Data Processing and Analysis: The central system helps gather, interpret, and decide on the information gathered from the subsystems.
4. Action and Feedback: According to the findings made above, the system can cause actions through actuators and provide feedback to the users.
5. User Interaction: Using interfaces, a user can provide commands to the system and/or receive information from the system.

IoT examples and real-world use cases

1. Smart Homes:

- Example: Smart thermostats such as Nest learn the user's preferences and habits and then automatically control the house's temperature.
- Application: Home automation is used to save energy, protect the house, and increase comfort.

2. Wearable Devices:

- Example: Wearable fitness devices, such as Fitbit, measure activity, heart rate, and sleep.
- Application: M-Health to monitor personal health and fitness.

3. Industrial IoT (IIoT):

- Example: Some advanced factory systems include predictive maintenance, where the health of equipment is monitored to predict failure.
- Application: Less operational downtime and improving on their efficiency.

4. Smart Cities:

- Example: Smart traffic systems change the traffic signals in real-time to ease congestion of the road.
- Application: Enhancing the physical facilities in urban areas and the overall welfare of the inhabitants.

- **Knowledge Check 1**

  Fill in the Blanks

  1. AI-powered chatbots improve ___________ and customer relations. (efficiency)
  2. AI and ML transform ___________ and decision-making procedures. (Automation)
  3. Internet of Things (IoT) links ___________ devices and allows the exchange of information. (connected)
  4. IoT applications can be extended from ___________ management to smart home gadgets. (Industrial)

- **Outcome-Based Activity 1**

  Identify new technologies like Artificial Intelligence, the Internet of Things, Cloud Computing, Blockchain, and Quantum Computing. Divide the applicants into groups and ask them to indicate possible applications, advantages and disadvantages of each technology. In what ways do these technologies determine the development of computing and alter the sectors?

16.3 Blockchain Technology

Introduction to Blockchain

Blockchain technology, which is now known as the technology behind cryptocurrencies, has been considered revolutionary in the recent past. In essence, Blockchain can be described as a decentralised, distributed digital ledger technology that is used to create a new, improved, and more efficient method of recording information. Let's explore the key concepts and components of blockchain:

1. What is Blockchain?

- Blockchain is a growing database placed on a chain of blocks, each containing a list of transactions.

- These blocks are connected by cryptographic algorithms that provide a chronological and secure chain of transactions.

2. Decentralisation

- While data in conventional centralised databases is managed and owned by a single central authority, blockchain works on a distributed network of nodes.
- Every node has a copy of the blockchain to prevent the system from having a single point of failure and also to make data immune to censorship and manipulation.

3. Transparency and Immutability

- The activities performed as a result of the transactions captured on the blockchain are available to all network participants,
- increasing the level of integrity.
- Once a transaction is stored, it cannot be changed or removed from the block – which is the case in blockchain since it can only be changed through consensus among the nodes within the network.

4. Consensus Mechanisms

- The consensus mechanism can be described as the procedures that help all the nodes in the network agree on the validity of the transaction and the order in which the transactions should be recorded.
- Some of the commonly known consensus mechanisms are the Proof of Work (PoW), Proof of Stake (PoS), and Delegated Proof of Stake (DPoS).

5. Cryptographic Security

- Blockchain applies encryption through cryptographic algorithms to make transactions safe and avoid forgery.
- In this case, public and private key pairs are used in user authentication and digital signatures.

6. Use Cases

- Blockchain technology is versatile and can be used in various sectors, such as finance, supply chain, health, property, and many more.
- Some of the areas where blockchain is applied are Cryptocurrency, Identity management, Financial assets, Supply chain management, Voting, and many others.

16.4 Future Trends in Computing

The dynamics of computing are determined by the progression of technologies that have been accelerated by the need for better-performing, efficient, and smarter systems. Here are some key trends that are expected to define the future of computing:

1. Quantum Computing

Quantum computing is said to disrupt several industries through the computational analysis of problems at a very fast rate. Quantum computers are dissimilar to the classical computers that apply bits (0's and 1's) and differ from them with qubits, which can be in more than one state at a time. This allows them to solve problems that are beyond the capability of current classical computers, such as big simulations and cryptography crackers. Major tech giants such as IBM, Google, and Microsoft are already on the path of creating realistic quantum computers.

2. AI and Machine Intelligence

Artificial intelligence and machine learning are increasingly technologies that are embedded in computing, allowing systems to learn from data and reason. Such technologies as deep learning, natural language processing, and reinforcement learning are now pushing development in fields such as self-driving cars, individualised healthcare, and smart assistants. Increasing its compatibility with other emerging technologies like IoT and edge computing will only increase the capabilities and possibilities of AI.

3. Edge Computing

Edge computing involves performing computations and data storage in the vicinity of the results required, hence minimising latencies and bandwidth consumption. This is especially so for real-time applications such as self-driving cars, factories, and smart cities. Edge computing is important in processing data locally and has the advantages of increasing efficiency and security as well as providing sufficient infrastructure for the increasing number of IoT devices.

4. Blockchain and Decentralised Technologies

The phenomenon of blockchain is revolutionising the process of data storage and performing operations with information through decentralisation. Apart from cryptocurrencies, the blockchain is used in supply chain solutions, identity verification, and smart contracts. These are decentralised technologies that provide attributes such as transparency, security, and efficiency in various industries.

5. Biocomputing

A similar concept with biocomputing is the ability to use biological molecules like DNA to carry out a computation. Although still in its infancy, this field has the potential to develop very efficient and parallel computing systems. Such uses may include handling big data, developing drugs, and bioengineering. It is suggested that biocomputing could present new ways of forming and processing information and problems.

16.5 Quantum Computing

Introduction to Quantum Computing

Quantum computing is a revolutionary technology based on the laws of quantum mechanics. It uses quantum bits, or qubits, to store and manipulate data in quantum states, solving problems that would be impossible for classical computers to solve. In contrast to classical computers, which employ bits as the elements of information, quantum computers use quantum bits, or qubits, to store and manipulate data in quantum states. Let's explore the key concepts, examples, applications, and challenges of quantum computing.

Key Concepts

1. Superposition

It gives qubits a state of superposition, where they can represent 0 and 1 simultaneously, making quantum computers compute different values at once.

2. Entanglement

They are quantum interactions and co-dependencies that occur between two or more qubits irrespective of their physical distances. This property allows quantum computers to carry out highly intertwined computations efficiently.

3. Quantum Gates

A quantum gate is similar to a classical gate and is defined as the basic unit of a quantum circuit. They selectively change the quantum state of qubits for operations such as superposition, entanglement, and measurement.

4. Quantum Algorithms

Quantum algorithms refer to algorithms that are devised to solve certain computational problems with the help of quantum mechanical characteristics that are superior to those of classical algorithms. For instance, Shor’s algorithm is used in the field of integer factorisation, and Grover’s algorithm is used for the unstructured search.

Examples and Applications

1. Cryptography

A quantum computer can attack RSA and ECC encryption techniques because Shor's algorithm can factor large numbers with great efficiency. On the other hand, post-quantum cryptographic algorithms like lattice-based cryptography secure future communication systems.

2. Optimisation

Optimisation problems can also be solved more efficiently using quantum computers than classical computers with the help of quantum algorithms such as the Quantum Approximate Optimisation Algorithm (QAOA). Some of the uses include asset allocation, procurement and supply chain management, and delivery networks.

3. Drug Discovery

Current quantum computers can emulate the molecule's formation and behaviour much faster than other forms of computers, enabling speedier drug discovery. Modern quantum algorithms such as VQE and QCS are used to model chemical reactions and screen candidates for drug discovery.

Challenges

1. Hardware Constraints

Physical implementation and growth of quantum hardware with enough coherence of qubits, connections between them, and errors still pose a challenge. Modern quantum processors suffer from noise, decoherence, and a small number of qubits, which limits the applicability of quantum algorithms.

2. Quantum Error Correction

Quantum error correction requires a greater number of qubits and more complex error correction codes, which leads to higher overhead and more resources needed for quantum computation. The ability to correct errors is critical as well to ensure that the calculations are accurate without requiring heftier hardware.

3. Algorithm Development

The development and enhancement of quantum algorithms for practical purposes involves knowledge of quantum mechanics, computer science, and the domain of application. The challenge of creating quantum algorithms that solve problems faster than classical algorithms for a vast number of issues is yet to be achieved.

- **Knowledge Check 2**

  Fill in the Blanks

1. Some future trends in computing comprise ___________ computing and edge computing. (cloud)
2. Quantum computing has the potential to solve complex problems and identify novel ___________ and cyphers. (algorithms)
3. Blockchain technology guarantees ___________ and decentralised transaction integrity. (security)
4. Algorithms used in quantum computing can ___________ problems in science and finance. (modernise)

- **Outcome-Based Activity 2**

  Discuss the tendencies of the further evolution of computing with the participation of artificial intelligence, edge computing, serverless structure, and the use of technology for the improvement of human life.

16.6 Summary

- Learned about Artificial Intelligence and Machine Learning, their working mechanism, how algorithms learn from the data and make predictions or decisions, and how these technologies are used in different fields.
- Exploring IoT as a concept that focuses on connecting different physical gadgets and facilitating data sharing over the internet and its use in smart cities, health, agriculture and industry.
- Explained the cloud computing service models, including IaaS, PaaS, and SaaS, as well as the deployment models, which are public, private, and hybrid cloud. Understand the advantages of cloud computing, such as scalability, flexibility, and cost.
- Based on blockchain, understood that it is not just an appendage of cryptocurrencies but also supply chains, digital signatures, and contracts.
- Learned the basics of quantum computing and how it can solve certain problems exponentially faster than classical computers, with uses in cryptography, optimisation, and simulation.
- Touched upon edge computing as a concept and its role in computing involving the processing of data closer to the point of origin, increasing efficiency and decreasing factors such as latency and bandwidth consumption, as well as integral real-world applications such as the IoT and self-driving automobiles.

- This paper also discussed serverless computing models, in which the cloud providers' premises manage the infrastructure, but the developers are only concerned with writing and deploying their code. It also discussed the advantages of this model, including scalability and cost efficiency.
- Explained the use of combining AI and ML to software development lifecycle, creating code, testing, and deployment tools and its capability of promoting productivity.

16.7 Keywords

- **Artificial intelligence:** Artificial intelligence, or AI, is a branch of computer science that deals with the development of intelligent machines that work like humans.
- **Machine Learning:** Machine Learning (ML) is a category of AI that employs algorithms and statistical models to allow computers to learn from experience and perform a given task efficiently.
- **Blockchain:** Blockchain is a distributed digital database made of a chain of blocks, each block containing a record of transactions.

16.8 Self-Assessment Questions

1. Examine the areas where AI and ML are likely to be used in various industries.
2. Discuss the meaning of the Internet of Things (IoT) and the prospects of interconnected items and smart networks.
3. What are some of the other applications of blockchain apart from its use in cryptocurrencies?
4. Get your ideas behind the trends that would shape the future of computing, like edge computing, serverless architecture, quantum computing
5. What are some challenges associated with IoT adoption?

16.9 References / Reference Reading

- Jain, V.K. *Machine Learning*. Khanna Publishing House, 2023.
- Chandra S.S., Vinod. *Artificial Intelligence & Machine Learning*. PHI Learning, 2023.

- Chopra, Rajiv. *Deep Learning: Fundamentals and Applications*. Sage Publications, 2023.
- Dhanalakshmi, S. *Blockchain Technology and Applications*. McGraw-Hill Education, 2023.