CBCS SCHEME

USN 1CR25EC190

1BESC104E

First Semester B.E./B.Tech. Degree Examination, Dec.2025/Jan.2026

Essentials of Information Technology

Time: 3 hrs. Max. Marks: 100

Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M : Marks , L: Bloom's level , C: Course outcomes.

			M	L	C
		Module – 1			
Q.1	**a.**	What is flip flop? Explain the working of a simple flip flop circuit.	**6**	**L2**	**CO1**
	b.	Explain the three major categories of machine instructions with suitable examples.	**6**	**L2**	**CO1**
	c.	What is machine cycle? Explain the procedure of computer programs are execution.	**8**	**L2**	**CO1**
		OR			
Q.2	**a.**	What is the role of Controllers while communicating computer with other devices.	**6**	**L2**	**CO1**
	b.	With block diagram explain the computer architecture.	**6**	**L3**	**CO1**
	c.	Explain the organization of magnetic systems, optical systems and flash drives for mass storage.	**8**	**L2**	**CO1**
		Module – 2			
Q.3	**a.**	What is bootstrapping? Explain the booting process.	**10**	**L3**	**CO2**
	b.	Explain the deadlock problem which arises during resource allocation.	**6**	**L2**	**CO2**
	c.	List the problem-solving phases used for program development.	**4**	**L1**	**CO2**
		OR			
Q.4	**a.**	Explain briefly the function of : i) Window manager ii) File Manager iii) Memory Manager iv) Scheduler	**8**	**L3**	**CO2**
	b.	Summarize the distinctions between a process, an algorithm and a program.	**6**	**L1**	**CO2**
	c.	Explain how multiprogramming works between various processes in the system.	**6**	**L2**	**CO2**
		Module – 3			
Q.5	**a.**	With neat diagram, explain Internet Architecture.	**10**	**L3**	**CO2**
	b.	Define Artificial Intelligence (AI) and discuss the ethical risk of biases in AI model in critical areas.	**4**	**L1**	**CO2**
	c.	What is URL? With diagram explain the segments of typical URL.	**6**	**L2**	**CO3**

		OR			
Q.6	**a.**	With suitable diagram explain the use of switch and repeater in networking.	**8**	**L3**	**CO2**
	b.	Explain the practices used to keep password safe and secure as part of good cyber hygiene.	**6**	**L3**	**CO5**
	c.	Define online tracking and explain the effects of social media addictions.	**6**	**L2**	**CO5**
		Module – 4			
Q.7	**a.**	Explain the traditional development phases of the software life cycle.	**8**	**L2**	**CO4**
	b.	Describe the concept of Modularity and explain why it is essential for large-scale software systems.	**6**	**L2**	**CO4**
	c.	With diagram explain the conceptual layers of database implementation.	**6**	**L2**	**CO4**
		OR			
Q.8	**a.**	Compare and contrast the Waterfall model with Incremental and Iterative software development methodologies.	**8**	**L3**	**CO3**
	b.	Explain the role of Unified Modeling Language (UML) in software engineering. Develop use case diagram for Hospital Records System.	**6**	**L2**	**CO4**
	c.	Explain with example Three relational database operations.	**6**	**L2**	**CO4**
		Module – 5			
Q.9	**a.**	Explain the basic structure of an HTML page with example.	**6**	**L2**	**CO3**
	b.	Explain the fundamental difference between 2D and 3D computer graphics. Which of these typically requires more computational power and why?	**6**	**L2**	**CO3**
	c.	Illustrate the importance of CSS in modern Web design. Develop a CSS to have page background color 'white', Heading-1 with color 'Black' with text alignment to center and paragraph with font family 'Times New Roman' with font size of 30px.	**8**	**L3**	**CO3**
		OR			
Q.10	**a.**	Explain how user can create tables in HTML to help organize data.	**6**	**L2**	**CO3**
	b.	Discuss the role of "Shading" and "Lighting" in the rendering process. How do they contribute to the realism of an image?	**6**	**L2**	**CO3**
	c.	Explain how the folder structure helps in linking files such as images or style sheets in HTML.	**8**	**L2**	**CO3**

* * * * *

MODULE-1

Q1 (a) What is a Flip-Flop? Explain the working of a simple flip-flop circuit.

Definition:

A **Flip-Flop** is a basic sequential digital circuit used to store **one bit (0 or 1)** of data. It has two stable states and is therefore called a **bistable device**.

Simple SR Flip-Flop:

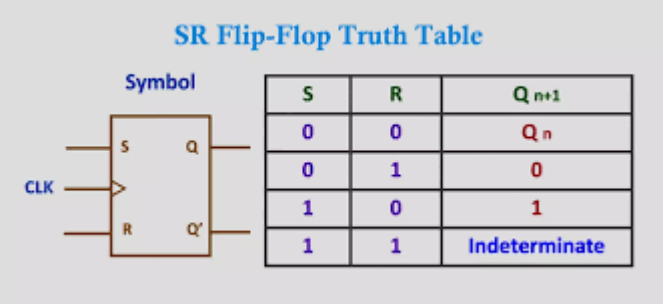
It is constructed using two NOR gates (or NAND gates).

Inputs:

- S → Set
- R → Reset

Outputs:

- Q
- $\overline{Q}$ (complement of Q)



Explanation:

- When **S = 1**, the flip-flop sets the output Q to 1.
- When **R = 1**, it resets Q to 0.
- When both inputs are 0, the previous value is retained.
- When both inputs are 1, it leads to an invalid condition.

Applications:

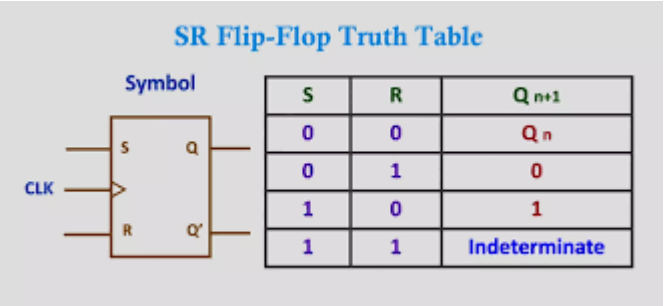
- Memory registers
- Counters
- Frequency dividers
- Data storage elements

Q1 (b) Explain the three major categories of machine instructions with suitable examples.

Machine instructions are commands given to the CPU to perform operations. They are mainly classified into three categories:

---

1. Data Transfer Instructions



Used to transfer data between registers, memory, or I/O devices.
Examples:

- MOV A, B → Move contents of B into A
- LOAD R1, 5000 → Load memory location 5000 into R1
- STORE R1, 6000

---

2. Arithmetic and Logical Instructions
Used to perform mathematical and logical operations.
Examples:

- ADD A, B → Add B to A
- SUB A, B → Subtract B from A
- AND A, B → Logical AND
- OR A, B

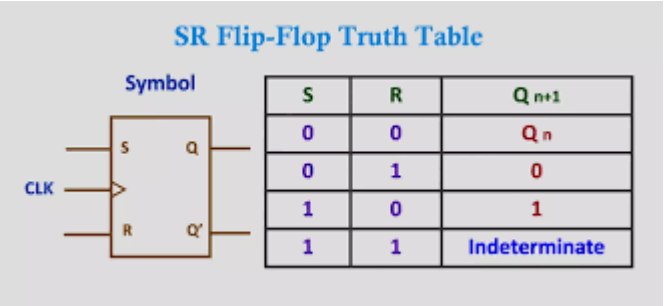
---

3. Control (Branching) Instructions
Used to control the flow of execution of a program.
Examples:

- JMP 2000 → Jump to memory location 2000
- CALL SUB → Call a subroutine
- RET → Return from subroutine

Q1 (c) What is Machine Cycle? Explain the procedure of program execution.
A **Machine Cycle** is the sequence of operations performed by the CPU to execute one instruction.
It consists of four stages:
1. Fetch
The CPU fetches the instruction from memory.
The address of the instruction is stored in the Program Counter (PC).
2. Decode
The Control Unit decodes the instruction and determines the operation to be performed.
3. Execute
The ALU performs the required arithmetic or logical operation.
4. Store
The result is stored in memory or registers.
Program Execution Process:

1. Program is loaded into memory.
2. PC points to the first instruction.
3. CPU performs Fetch → Decode → Execute → Store.



4. PC increments to the next instruction.
5. The cycle repeats until the program ends.

Q2 (a) What is the role of Controllers while communicating computer with other devices?

A **Controller** is a hardware device that manages communication between the CPU and peripheral devices.

Role of Controllers:

1. Acts as an interface between CPU and devices.
2. Converts data formats between system and device.
3. Sends and receives control signals.
4. Provides buffering to handle speed differences.
5. Detects and reports transmission errors.

Examples:

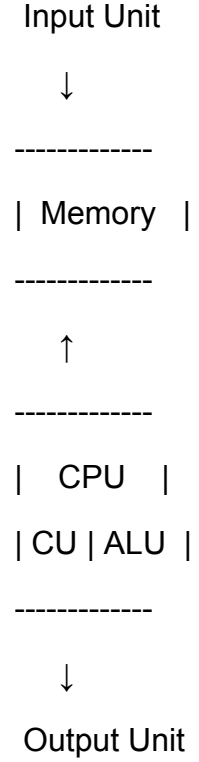
- Disk Controller
- USB Controller
- Display Controller

Q2 (b) With block diagram explain the computer architecture.

Computer architecture consists of the following main components:

1. Input Unit
2. Output Unit
3. Memory Unit
4. Central Processing Unit (CPU)
    - Control Unit (CU)

- Arithmetic Logic Unit (ALU)
- Registers



Explanation:

- **Input Unit**: Accepts data and instructions.
- **Memory Unit**: Stores data and instructions.
- **Control Unit (CU)**: Directs all operations of the computer.
- **ALU**: Performs arithmetic and logical operations.
- **Output Unit**: Displays processed results.

Q2 (c) Explain the organization of magnetic systems, optical systems and flash drives for mass storage.

Mass storage devices are used to store large amounts of data permanently.

---

1. Magnetic Storage Systems

Examples: Hard Disk, Magnetic Tape

- Data stored using magnetic coating.
- Organized into Tracks, Sectors, and Cylinders.
- Read/Write head magnetizes spots to store binary data.
- Has moving mechanical parts.

---

2. Optical Storage Systems

Examples: CD, DVD, Blu-ray

- Data stored in spiral tracks.
- Uses laser beam for reading and writing.
- Data represented as pits and lands.
- Moderate speed.

---

3. Flash Storage Systems

Examples: Pen drive, SSD

- Uses NAND flash memory.
- Stores data using electrical charge.
- No moving parts.
- Faster and more reliable.

MODULE-2

Q3 (a) What is Bootstrapping? Explain the booting process.

Bootstrapping:

Bootstrapping is the process of loading the operating system into the main memory when the computer is switched on.

It is also called the **booting process**.

Booting Process:

1. **Power On**
   - When the computer is switched on, power is supplied to all components.
2. **POST (Power-On Self Test)**
   - BIOS checks hardware components such as RAM, keyboard, hard disk, etc.
   - If any error occurs, it displays an error message.
3. **Loading Boot Loader**
   - BIOS searches for a bootable device (HDD/SSD/USB).
   - It loads the boot loader into RAM.
4. **Loading Operating System**
   - Boot loader loads the operating system kernel into memory.
5. **System Initialization**
   - OS initializes drivers and system services.
   - System becomes ready for user interaction.

Types of Booting:

- Cold Booting – Starting computer from power OFF state.
- Warm Booting – Restarting the system without turning power OFF.

Q3 (b) Explain the deadlock problem which arises during resource allocation.

A **deadlock** is a situation in which two or more processes are unable to proceed because each is waiting for a resource held by another process.

Example:

- Process P1 holds Resource R1 and waits for R2.
- Process P2 holds Resource R2 and waits for R1.
- Neither process can continue.

Four Necessary Conditions for Deadlock:

1. **Mutual Exclusion**
    - Only one process can use a resource at a time.
2. **Hold and Wait**
    - A process holding a resource waits for another.
3. **No Preemption**
    - Resources cannot be forcibly taken away.
4. **Circular Wait**
    - Processes form a circular chain waiting for resources.

Methods to Handle Deadlock:

- Deadlock Prevention
- Deadlock Avoidance
- Deadlock Detection and Recovery

Q3 (c) List the problem-solving phases used for program development.

The main phases in program development are:

1. **Problem Definition**
    - Clearly understand and define the problem.
2. **Analysis**
    - Identify inputs, outputs, and constraints.
3. **Algorithm Design**
    - Develop step-by-step solution.
4. **Flowchart / Pseudocode**
    - Represent logic diagrammatically.
5. **Coding**
    - Write the program in a programming language.
6. **Compilation and Execution**
    - Translate and run the program.
7. **Testing and Debugging**
    - Remove errors.
8. **Documentation**
    - Prepare user and technical documents.
9. **Maintenance**
    - Modify and update the program when required.

Q4 (a) Explain briefly the function of:

i) Window Manager
ii) File Manager
iii) Memory Manager
iv) Scheduler

These are important components of an Operating System.

i) Window Manager

The Window Manager is responsible for managing graphical user interface (GUI) windows.

Functions:

1. Creates and displays windows on the screen.
2. Manages window operations such as open, close, minimize, maximize, and resize.
3. Handles overlapping of windows.
4. Manages user interactions like mouse clicks and keyboard input.
5. Controls the layout and appearance of windows.
6. Maintains focus between active and inactive windows.

It ensures smooth graphical interaction between the user and the system.

ii) File Manager

The File Manager handles file system organization and file operations.

Functions:

1. Creates and deletes files and folders.
2. Provides naming and directory structure.
3. Supports copy, move, rename operations.
4. Maintains file permissions and security.
5. Keeps track of file locations on storage devices.
6. Organizes files using hierarchical directory structure.

It enables systematic storage and retrieval of data.

iii) Memory Manager

The Memory Manager controls and coordinates main memory usage.

Functions:

1. Allocates memory to processes.
2. Deallocates memory after process completion.
3. Keeps track of used and free memory space.
4. Supports virtual memory management.
5. Prevents one process from accessing another's memory.
6. Manages paging and segmentation.

It ensures efficient and safe utilization of RAM.

iv) Scheduler

The Scheduler decides which process gets CPU time.

Functions:

1. Selects the next process for execution.
2. Maintains ready queue of processes.
3. Allocates CPU based on scheduling algorithm.
4. Performs context switching.
5. Ensures fair distribution of CPU time.
6. Improves system responsiveness.

Types of scheduling:

- First Come First Serve (FCFS)
- Shortest Job First (SJF)
- Round Robin
- Priority Scheduling

Q4 (b) Summarize the distinctions between a process, an algorithm and a program.

Algorithm

- A step-by-step logical procedure to solve a problem.
- Written in simple English or pseudocode.
- Independent of programming language.
- Not directly executable by computer.

Example: Steps to calculate factorial.

Program

- A set of instructions written in a programming language.
- Implements an algorithm.
- Stored on disk.
- Needs a compiler/interpreter to execute.

Example: C or Python code for factorial.

Process

- A program that is currently executing.
- Active entity.
- Has its own memory space, registers, and resources.
- Managed by operating system.

Example: Running browser or text editor.

Feature	Algorithm	Program	Process
Nature	Logical steps	Written code	Executing program
Executable	No	Yes (after compilation)	Yes
State	Static	Static	Dynamic
Managed by OS	No	No	Yes

Q4 (c) Explain how multiprogramming works between various processes in the system.

Multiprogramming is a technique where multiple programs are loaded into memory at the same time to increase CPU utilization.

Working of Multiprogramming:

1. Several processes are loaded into main memory.
2. CPU executes one process at a time.
3. When a process performs I/O operation:
    - It enters a waiting state.
4. CPU switches to another ready process.
5. Context switching saves the current process state.
6. This continues until all processes complete.

Process States in Multiprogramming:

- New
- Ready
- Running
- Waiting
- Terminated

Advantages:

1. Better CPU utilization.
2. Reduced idle time.

3. Increased throughput.
4. Efficient resource usage.

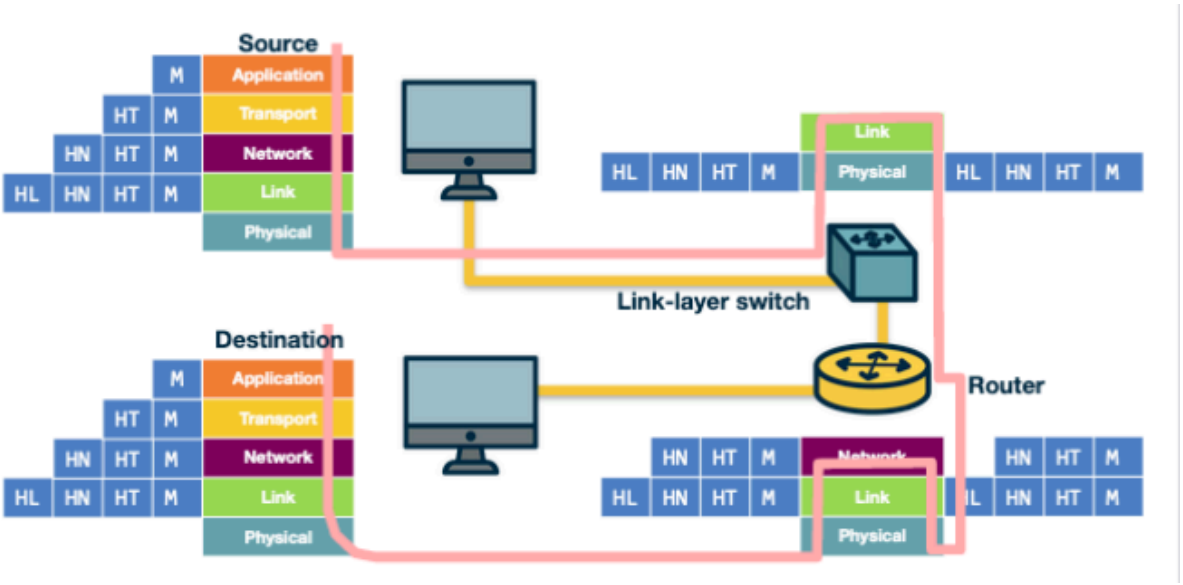
MODULE-3

Q.5 (a) With neat diagram, explain Internet Architecture. (10 Marks)

Internet Architecture

The Internet follows a **layered architecture model**, primarily based on the **TCP/IP model**. It is a global network of interconnected networks that communicate using standardized protocols.

Neat Diagram –



Explanation of Each Layer

Application Layer

- Provides services to end users.
- Handles web browsing, email, file transfer.
- Protocols:
    - **HTTP/HTTPS** – Web communication
    - **FTP** – File transfer
    - **SMTP/POP3/IMAP** – Email
    - **DNS** – Domain name resolution

Transport Layer

- Responsible for **end-to-end communication**.
- Ensures reliable or fast data delivery.
- Protocols:
    - **TCP (Transmission Control Protocol)**
        - Reliable
        - Error checking
        - Flow control
    - **UDP (User Datagram Protocol)**
        - Faster
        - No guarantee of delivery

Internet Layer

- Responsible for **logical addressing and routing**.

- Uses **IP addresses**.
- Determines best path to destination.
- Protocols:
    - IP
    - ICMP
    - ARP

Network Access Layer

- Handles **physical transmission of data**.
- Concerned with hardware, cables, MAC addressing.
- Examples:
    - Ethernet
    - Wi-Fi
    - Fiber optics

How Data Flows in Internet Architecture

1. Application creates data.
2. The transport layer segments it.
3. The Internet layer adds an IP address.
4. The network layer sends it physically.
5. The reverse process happens at the receiver.

Internet architecture follows a **layered, protocol-based communication model** ensuring scalability, interoperability, and reliable global communication.

Q.5 (b) Define Artificial Intelligence (AI) and discuss ethical risk of biases in AI models in critical areas. (4 Marks)

Definition of Artificial Intelligence

Artificial Intelligence (AI) is a branch of computer science that enables machines to simulate human intelligence, including learning, reasoning, problem-solving, and decision-making.

AI systems use:

- Machine Learning
- Deep Learning
- Natural Language Processing
- Computer Vision

Ethical Risks of Bias in AI (Critical Areas)

Bias occurs when AI systems produce unfair or discriminatory outcomes due to biased training data or flawed design.

Healthcare

- Biased diagnostic models may misdiagnose certain ethnic groups.
- Can lead to incorrect treatment decisions.

Criminal Justice

- Predictive policing tools may unfairly target specific communities.
- Risk assessment tools may show racial bias.

Recruitment

- AI hiring systems may prefer certain genders or backgrounds.
- Discrimination in job selection.

Finance

- Biased loan approval systems.
- Credit scoring discrimination.

Bias in AI can cause serious ethical, social, and legal consequences. Therefore:

- Fair datasets
- Transparent models
- Regular auditing
- Human oversight
  are essential in critical applications.

Q.5 (c) What is URL? With diagram explain the segments of typical URL. (6 Marks)

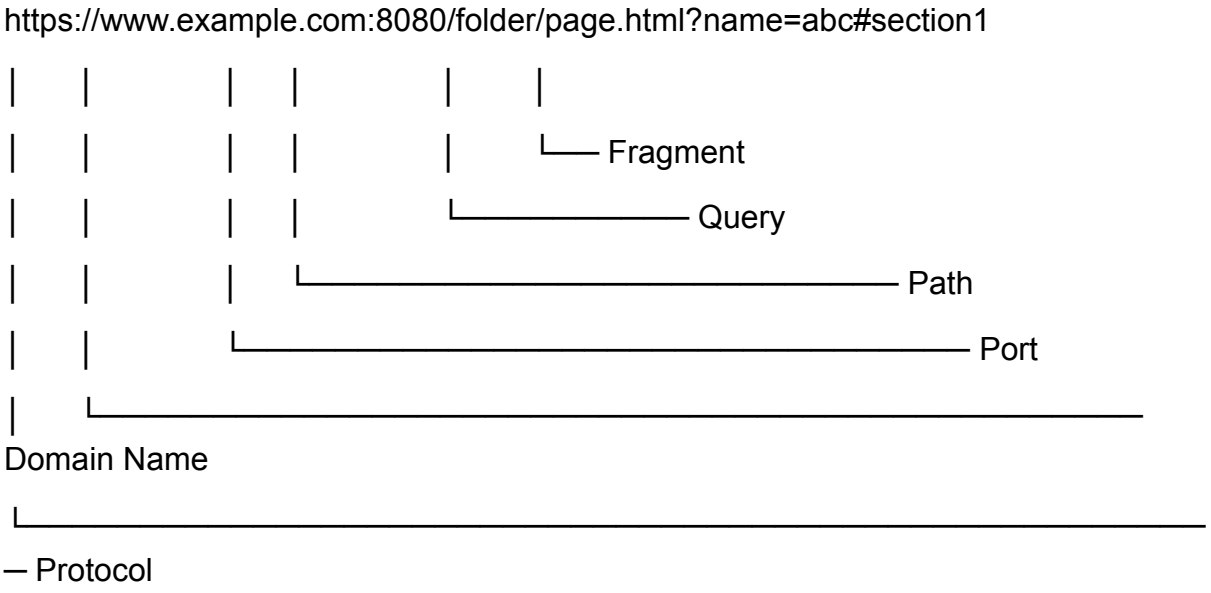
Definition of URL

URL (Uniform Resource Locator) is the address used to locate a resource on the Internet.

Example:

https://www.example.com:8080/folder/page.html?name=abc#section1

Diagram of URL Structure



Segments Explanation

Protocol (Scheme)

- `https`
- Defines communication method.
- Examples: HTTP, HTTPS, FTP

Domain Name

- `www.example.com`
- Identifies the server.
- Converted to IP using DNS.

Port (Optional)

- `8080`

- Communication endpoint.
- Default:
    - HTTP → 80
    - HTTPS → 443

Path

- `/folder/page.html`
- Location of resource in server.

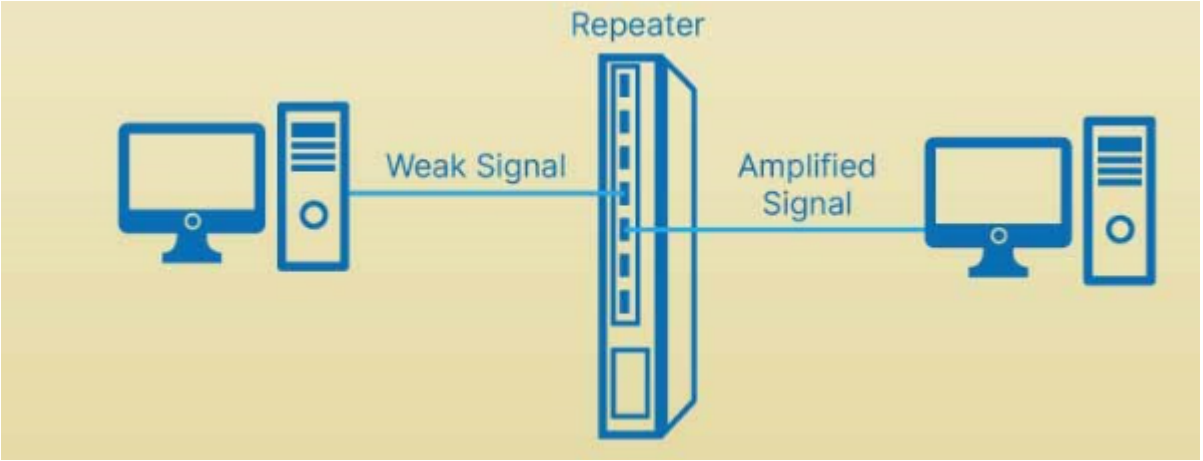
Query String

- `?name=abc`
- Sends data to server.

Fragment

- `#section1`
- Refers to a specific part of the webpage.

A URL uniquely identifies and locates resources on the Internet by combining protocol, domain, and resource path information.



Q.6 (a) With suitable diagrams explain the use of Switch and Repeater in networking.

Repeater

A Repeater is a Layer 1 (Physical Layer) device used to regenerate and amplify weak signals in a network.

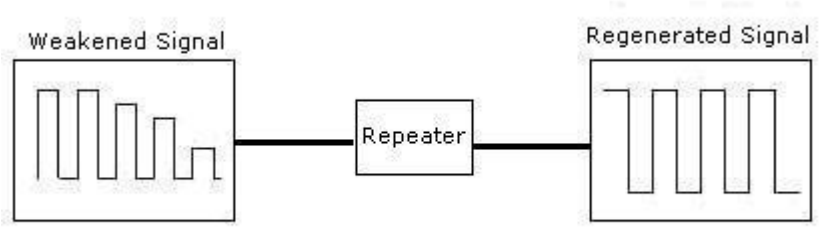
Purpose

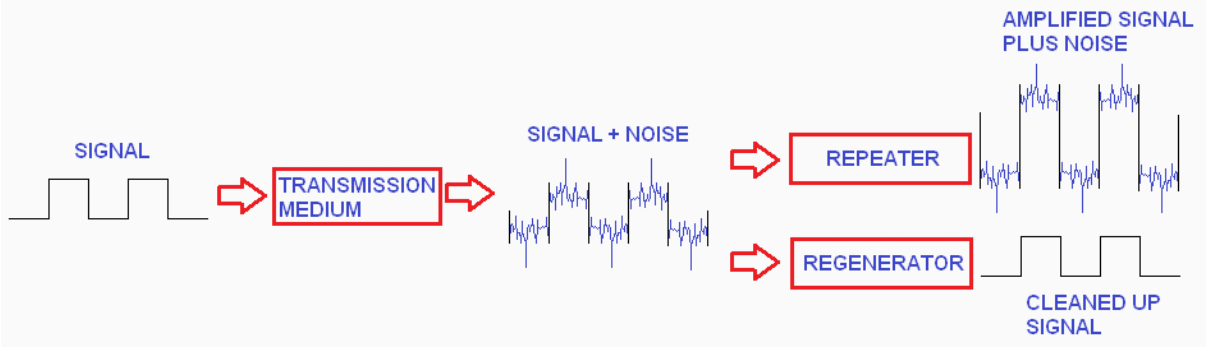
- Extends network distance
- Restores attenuated signals
- Reduces signal distortion

Working Principle

- Receives weak signal
- Regenerates (reshapes + amplifies)
- Retransmits to next segment

Diagram





Switch

A Switch is a Layer 2 (Data Link Layer) device that forwards frames based on MAC addresses.

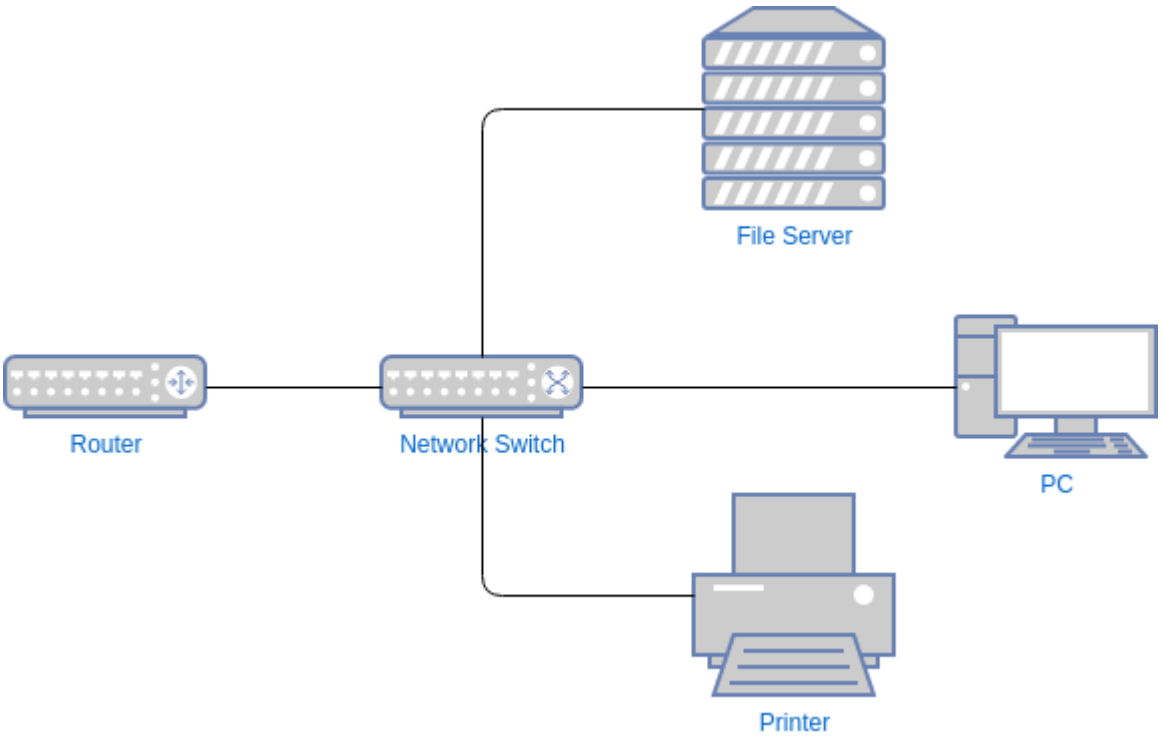
Purpose

- Connects multiple devices in LAN
- Reduces collisions
- Improves bandwidth efficiency

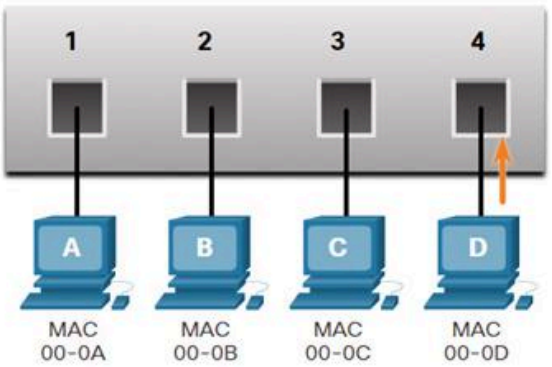
Working Principle

- Maintains MAC address table
- Forwards frames only to destination port
- Each port is a separate collision domain

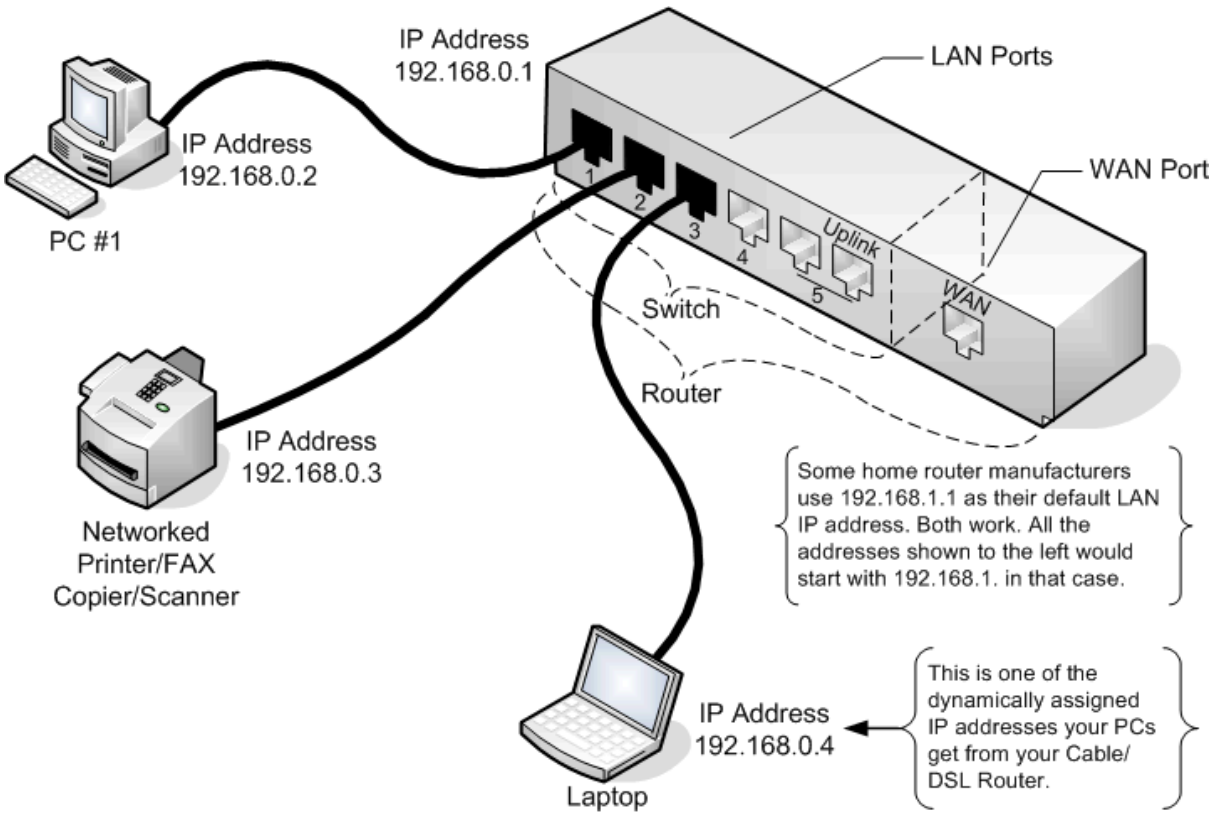
Diagram



MAC Address Table	
Port	**MAC Address**
1	00-0A
4	00-0D



The switch adds the port number and MAC address for PC-D to its MAC address table.



Used in offices, labs, and data centers to connect computers, printers, and servers.

Key Points

- Works at Data Link Layer
- Uses MAC address
- Reduces collisions
- More intelligent than hub/repeater

Difference Between Repeater and Switch

Feature	Repeater	Switch
OSI Layer	Layer 1	Layer 2
Signal Processing	Regenerates signal	Forwards frames
Address Awareness	No	Yes (MAC)

Collision Control	No	Yes
Intelligence Level	Very Low	Moderate

Q.6 (b) Explain the practices used to keep password safe and secure as part of good cyber hygiene. (6 Marks)

Cyber hygiene refers to practices adopted to maintain system security and protect digital information.

Best Practices for Password Security

Use Strong Passwords

- Minimum 12–16 characters
- Mix of uppercase, lowercase, numbers, symbols
- Avoid dictionary words

Avoid Personal Information

- No DOB, phone number, name
- Hard to guess credentials

Use Unique Passwords

- Different password for each account
- Prevents credential stuffing attacks

Enable Multi-Factor Authentication (MFA)

- OTP, authenticator app, biometric

- Adds extra security layer

Use Password Manager

- Secure storage of passwords
- Generates strong passwords

Change Passwords Periodically

- Especially after suspected breach

Do Not Share Passwords

- Avoid sending via email or messages

Beware of Phishing

- Do not enter credentials on fake websites

Q.6 (c) Define online tracking and explain the effects of social media addictions. (6 Marks)

Online Tracking – Definition

Online tracking refers to the collection of user data and behavior on the internet using:

- Cookies
- Tracking pixels
- Browser fingerprinting
- IP address logging

Purpose:

- Targeted advertising
- User behavior analysis
- Personalization

Effects of Social Media Addiction

Mental Health Issues

- Anxiety
- Depression
- Low self-esteem

Reduced Productivity

- Distraction from studies/work
- Time mismanagement

Sleep Disturbance

- Late-night scrolling
- Blue light exposure

Social Isolation

- Reduced face-to-face interaction

Cyberbullying Risk

- Emotional trauma

Privacy Risks

- Oversharing personal data

Online tracking raises privacy concerns, and excessive social media use negatively impacts mental health, productivity, and personal relationships.

Q7 (a) Explain the traditional development phases of the Software Life Cycle. (8 Marks)

The **traditional Software Development Life Cycle (SDLC)** generally refers to the **Waterfall Model**, where development proceeds in sequential phases. Each phase must be completed before moving to the next.

1. Requirement Analysis

- Collect and analyze user requirements.
- Identify functional and non-functional requirements.
- Output: **Software Requirement Specification (SRS)** document.

2. System Design

- Transform requirements into system architecture.
- High-Level Design (HLD): overall architecture.
- Low-Level Design (LLD): module-level design.
- Output: Design documents.

3. Implementation (Coding)

- Developers write code according to design specifications.
- Each module is developed independently.
- Output: Source code.

4. Testing

- Verify and validate the system.
- Types:
    - Unit Testing
    - Integration Testing
    - System Testing
    - Acceptance Testing
- Objective: Detect and fix defects.

5. Deployment

- The system is delivered and installed in the user environment.
- User training and documentation provided.

6. Maintenance

- Post-deployment support.
- Types of maintenance:
    - Corrective (bug fixing)
    - Adaptive (environment changes)
    - Perfective (performance improvements)
    - Preventive (future issue prevention)

Diagram (Sequential Flow):

Requirement → Design → Coding → Testing → Deployment → Maintenance

Q7 (b) Describe the concept of Modularity and explain why it is essential for large-scale software systems. (6 Marks)

Modularity is the design principle of dividing a software system into smaller, independent, and manageable units called **modules**.

Each module:

- Performs a specific function
- Has high cohesion
- Has low coupling with other modules

Key Concepts:

- **High Cohesion** → Module performs one well-defined task.
- **Low Coupling** → Minimal dependency between modules.
- **Information Hiding** → Internal implementation is hidden.

Why Modularity is Essential for Large-Scale Systems:

1. **Manageability**
    - Large systems become easier to understand and develop.
2. **Parallel Development**
    - Multiple teams can work on different modules simultaneously.
3. **Maintainability**
    - Changes in one module do not heavily affect others.
4. **Reusability**
    - Modules can be reused in other projects.

5. **Scalability**
    - New features can be added without redesigning the entire system.
6. **Testing & Debugging**
    - Easier to test modules independently.

Modularity improves system quality, reduces complexity, and enhances maintainability, making it essential for enterprise-level systems.

Q7 (c) With diagrams explain the conceptual layers of database implementation. (6 Marks)

Database systems are implemented using a **three-level architecture (ANSI-SPARC architecture)**.

1. External Level (View Level)

- User-specific views of the database.
- Different users see different views.
- Hides complexity.

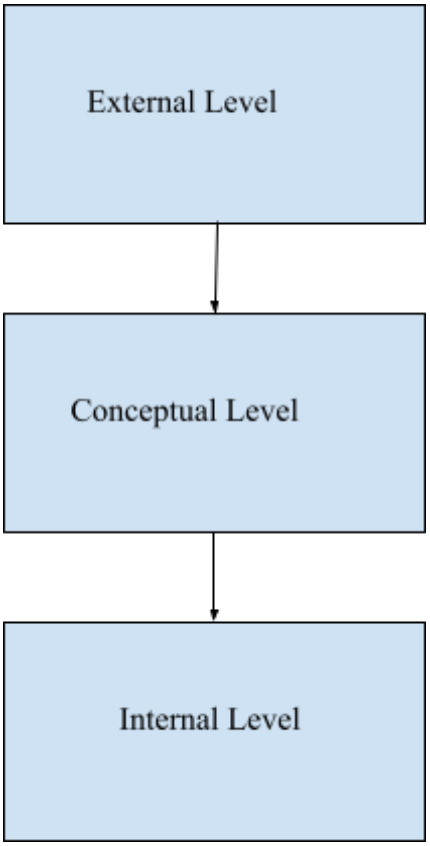
2. Conceptual Level (Logical Level)

- Global logical structure of the database.
- Defines entities, attributes, relationships.
- Independent of physical storage.

3. Internal Level (Physical Level)

- Describes how data is physically stored.
- File structures, indexing, storage allocation.

Diagram:



Advantages of Three-Level Architecture:

- **Data Abstraction**
- **Data Independence**
    - Logical Data Independence
    - Physical Data Independence
- **Security**
- **Multiple User Views**

Q8 (a) Compare and contrast the Waterfall model with Incremental and Iterative software development methodologies. (8 Marks)

1. Waterfall Model

A linear, sequential model where each phase must be completed before the next begins.

Phases:
Requirements → Design → Implementation → Testing → Deployment → Maintenance

2. Incremental Model

The system is developed and delivered in small functional increments.
Each increment adds new features to the previous release.

3. Iterative Model

The system is developed through repeated cycles (iterations).
Each iteration refines and improves the previous version.

Comparison Table

Feature	Waterfall	Incremental	Iterative
Process Flow	Linear & sequential	Delivered in increments	Repeated refinement cycles
Requirement Stability	Must be fixed at start	Can evolve per increment	Can change between iterations
Customer Feedback	After complete product	After each increment	After every iteration
Risk Handling	High risk	Reduced risk	Risk handled early

Delivery	Single final delivery	Multiple partial deliveries	Multiple improved versions
Flexibility	Low	Moderate	High
Testing	After coding phase	After each increment	In every iteration

Differences

- Waterfall is rigid and suitable for well-defined projects.
- Incremental focuses on delivering functional modules one by one.
- Iterative focuses on improving and refining the system progressively.
- Incremental & Iterative models support better change management compared to Waterfall.

Q8 (b) Explain the role of Unified Modeling Language (UML) in software engineering. Develop use case diagram for Hospital Records System. (6 Marks)

Role of UML in Software Engineering

Unified Modeling Language (UML) is a standardized modeling language used to visualize, specify, construct, and document software systems.

Importance:

1. Visual representation of system architecture.
2. Improves communication between stakeholders.
3. Helps in requirement clarification.

4. Supports object-oriented design.
5. Provides multiple diagram types (Use case, Class, Sequence, Activity, etc.).
6. Serves as a blueprint before coding.

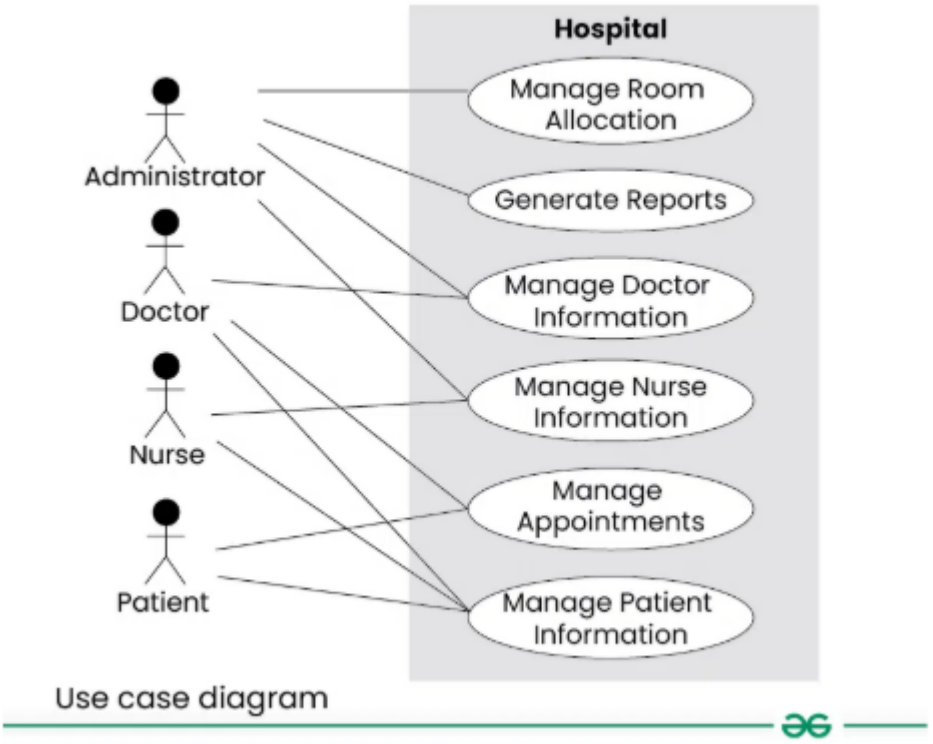
Use Case Diagram – Hospital Records System

Actors:

- Patient
- Doctor
- Receptionist
- Admin

Use Cases:

- Register Patient
- Book Appointment
- Maintain Medical Records
- View Medical History
- Generate Bill
- Manage Staff



Use case diagram

Q8 (c) Explain with example Three relational database operations. (6 Marks)

Relational algebra provides operations to manipulate data in relational databases.

1. Selection (σ)

- Selects rows that satisfy a given condition.
- Operates on tuples (rows).

Example:
σ (Department = 'CSE') (Student)

Meaning: Select students belonging to the CSE department.

2. Projection (π)

- Selects specific attributes (columns).
- Removes unwanted attributes.

Example:
π (Name, RollNo) (Student)

Meaning: Display only Name and RollNo from Student table.

3. Join (⋈)

- Combines tuples from two relations based on a related attribute.

Example:

Student (RollNo, Name, DeptID)
Department (DeptID, DeptName)

Student ⋈ Department

Result: Combines student details with department name using DeptID.

Q.9 (a) Explain the basic structure of an HTML page with examples. (6 Marks)

An HTML (HyperText Markup Language) document follows a standard structure that defines how content is organized and displayed in a web browser.

Basic Structure Components

1. `<!DOCTYPE html>`
    - Declares the document type and version (HTML5).
2. `<html>`
    - Root element that contains the entire HTML document.
3. `<head>`
    - Contains metadata (title, character set, CSS links, scripts).
    - Not directly visible on the webpage.

4. `<title>`
    - Defines the title shown in the browser tab.
5. `<body>`
    - Contains visible content like text, images, links, tables, etc.

Example HTML Page

```
<!DOCTYPE html>
<html>
<head>
  <title>My First Web Page</title>
</head>
<body>
  <h1>Welcome to HTML</h1>
  <p>This is a simple HTML page example.</p>
</body>
</html>
```

Explanation:

- The `<h1>` tag defines a main heading.
- The `<p>` tag defines a paragraph.
- Everything inside `<body>` is displayed in the browser.

Q.9 (b) Explain the fundamental difference between 2D and 3D computer graphics. Which typically requires more computational power and why? (6 Marks)

2D Computer Graphics

- Uses two dimensions: X and Y axes.
- No depth (Z-axis).
- Objects are flat.
- Used in:
    - Web graphics
    - UI design
    - 2D games
    - Logos and icons

Example: Vector images, bitmap images.

3D Computer Graphics

- Uses three dimensions: X, Y, and Z axes.
- Includes depth and perspective.
- Objects have volume and realism.
- Used in:
    - Video games
    - Animation
    - Simulation

- Virtual Reality

Fundamental Differences

Feature	2D Graphics	3D Graphics
Dimensions	X, Y	X, Y, Z
Depth	No depth	Has depth
Realism	Less realistic	Highly realistic
Rendering	Simple	Complex (lighting, shading, textures)
Processing	Lower	Higher

Which Requires More Computational Power?

3D graphics require more computational power because:

1. Complex geometric calculations.
2. Perspective transformation.
3. Lighting and shading calculations.
4. Texture mapping.

5. Rendering and frame buffering.

6. Real-time animation physics.

3D rendering involves matrix transformations and intensive GPU processing, making it computationally heavier than 2D graphics.

Q.9 (c) Illustrate the importance of CSS in modern Web design. Develop a CSS as specified. (8 Marks)

Importance of CSS (Cascading Style Sheets)

1. Separates content (HTML) from presentation.

2. Improves maintainability.

3. Enables responsive design.

4. Enhances consistency across pages.

5. Reduces code repetition.

6. Improves page loading speed.

Required Design Specifications

- Page background color: White
- Heading-1 color: Black
- Heading-1 text aligned: Center
- Paragraph font family: Times New Roman
- Paragraph font size: 30px

CSS Code

```
body {
```

```
    background-color: white;
}

h1 {
    color: black;
    text-align: center;
}

p {
    font-family: "Times New Roman", serif;
    font-size: 30px;
}
```

Example HTML with CSS

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS Example</title>
    <style>
        body {
            background-color: white;
        }

        h1 {
```

```
        color: black;
        text-align: center;
    }

    p {
        font-family: "Times New Roman", serif;
        font-size: 30px;
    }
  </style>
</head>
<body>
  <h1>Welcome</h1>
  <p>This paragraph uses Times New Roman with 30px size.</p>
</body>
</html>
```

Q.10 (a) Explain how user can create tables in HTML to help organize data. (6 Marks)

HTML tables are used to organize data in **rows and columns** format.

Basic Table Tags

1. `<table>` → Defines the table.
2. `<tr>` (Table Row) → Defines a row.
3. `<th>` (Table Header) → Defines header cell (bold & centered).

4. `<td>` (Table Data) → Defines normal data cell.
5. `<caption>` → Adds table title (optional).
6. `<thead>`, `<tbody>`, `<tfoot>` → Logical grouping (optional).

Basic Structure

```
<table border="1">
  <tr>
    <th>Name</th>
    <th>Department</th>
    <th>Marks</th>
  </tr>
  <tr>
    <td>Rahul</td>
    <td>CSE</td>
    <td>85</td>
  </tr>
  <tr>
    <td>Anita</td>
    <td>ECE</td>
    <td>90</td>
  </tr>
</table>
```

Explanation:

- `<table>` creates the table.
- `<tr>` creates rows.
- `<th>` creates headings.
- `<td>` inserts data.
- `border="1"` adds a visible border (basic styling).

Tables help in organizing:

- Student records
- Timetables
- Product lists
- Financial reports

Q.10 (b) Discuss the role of "Shading" and "Lighting" in the rendering process. How do they contribute to realism? (6 Marks)

In computer graphics, **rendering** is the process of generating a 2D image from a 3D model.

1. Lighting

Lighting determines how light interacts with objects in a scene.

Types of Lighting:

- Ambient lighting (general light)
- Diffuse lighting (scattered light)
- Specular lighting (highlight reflections)

Role:

- Defines brightness and visibility.
- Creates highlights and shadows.
- Shows surface properties.

Without lighting → Objects look flat and unrealistic.

2. Shading

Shading calculates how light intensity varies across surfaces.

Common Shading Techniques:

- Flat Shading
- Gouraud Shading
- Phong Shading

Role:

- Smoothens surfaces.
- Adds depth and curvature.
- Enhances surface realism.

Contribution to Realism

Lighting + Shading together:

- Create depth perception.
- Show texture and material.
- Simulate real-world physics.

- Improve 3D appearance.

More advanced rendering includes:

- Shadow mapping
- Ray tracing
- Global illumination

Thus, shading and lighting are essential for producing realistic 3D images.

Q.10 (c) Explain how the folder structure helps in linking files such as images or style sheets in HTML. (8 Marks)

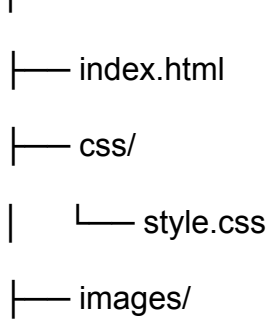
What is Folder Structure?

Folder structure organizes project files into directories such as:

- HTML files
- CSS files
- Images
- JavaScript files

Example Project Structure:

project/



```
│      └── logo.png
└── js/
    └── script.js
```

Linking CSS File

Inside `index.html`:

<link rel="stylesheet" href="css/style.css">

Explanation:

- `css/` → Folder name
- `style.css` → File inside that folder

Linking Image

<img src="images/logo.png" alt="Logo">

Types of File Paths

1. Relative Path
    - Based on current file location.
    - Example: `images/pic.jpg`
2. Absolute Path
    - Full URL.
    - Example: `https://example.com/images/pic.jpg`

3. Parent Directory
    - `../` → Move one folder up.
    - Example:
4. <link rel="stylesheet" href="../css/style.css">

Importance of Proper Folder Structure

1. Better project organization.
2. Easier maintenance.
3. Avoid broken links.
4. Faster development.
5. Scalability for large projects.

Proper folder structuring ensures correct file referencing and smooth website functioning.