SE Modul - 1

What is Program

. **Program** – it is a set of instructions

**Que:** **Write a simple "Hello World" program in two different programming languages of your choice. Compare the structure and syntax.?**

**Ans**. C language code for hello word

#include<stdio.h>

Void main()

{

Printf(“hello word”);

}

**Output: hello word**

* structure and syntax
* Requires a main() function and header file.
* Uses semicolons and braces.
* C is a compiled language, so it needs to
* Be copiled before running.
* python code for hello word

Print (“hello, word!”)

**Output:** hello word;

* Structure and Syntax
  + - very simple and clean
    - No need to define a main function
    - NO semicolons and headers
    - interpreted language: runs directly sours code

What is Programming

**. Programming** is the process of creating a set of instruction that tell a computer how to perform a specific task. These instructions are written using a **programming language**

**Que: Programming process**

1. Problem definition
2. Planning the solution
3. Coding
4. Testing and Debugging
5. Documentation
6. Maintenance

**Que: Type of Programming Languages.?**

1. Procedural programming Language

Ex – C Language

2. Object Oriented Programming

Ex – C++ Language

3. Logical Programming

Ex – Prolog Language

4.Function Programming

Ex – Python Language

**Que: Differences between High-level and low-level programming languages.?**

1. **High-level Languages:**

* **Abstraction:** They use syntax and structures closer to natural language,

Hiding the complexities of the underlying hardware.

* + **Readability:** Easier for programmers to understand, write and maintain code.
  + **Portability:** Code can be executed on different platforms with minimal or no Changes.
  + **Examples:** Python, java, C++, JavaScript,
  + **Uses:** Developing applications, websites and other software where ease of
    - Development and portability are prioritized.

**. Cons:** Can be slower and less memory-efficient than low-level languages.

1. **Low – Level Languages:**

* **Abstraction:** Close to machine code, requiring a deep understanding of

Computer architecture.

* **Readability:** Harder for humans to understand due to the use of mnemonics

and binary code.

* **Portability:** Generally not portable and tightly coupled with specific hardware or operating systems.
* **Examples:** Assembly languages, machine code.
* **Uses:** Developing system software, device drivers, and performance-critical

applications where speed and hardware control are crucial.

* **Cons:** Difficult to learn, program, and maintain.

**World Wide Web: WWW**

* WWWis a collection of webpages and resources that are accessed through the internet using web

Web browsers.

* The internet works as a global network of interconnected computers that communicate using standardized protocols like TCP/IP.

**Que: Research and create a diagram of how data is transmitted from a client to a server over the internet.?**

Ans. **1. Client Request:**

* You open a browser and type a URL (e.g. [www.example.com](http://www.example.com)).
* the browser a sends a request to get at web pages.

**2. DNS Resolution:**

- The browsers asks the DNS Server to translate to domain name (e.g. [www.example.com](http://www.example.com)) into a

IP address.

**3. Sending Request:**

- the client sends an HTTP/HTTPS request to the server using the established connection.

**4. Server receives the Request:**

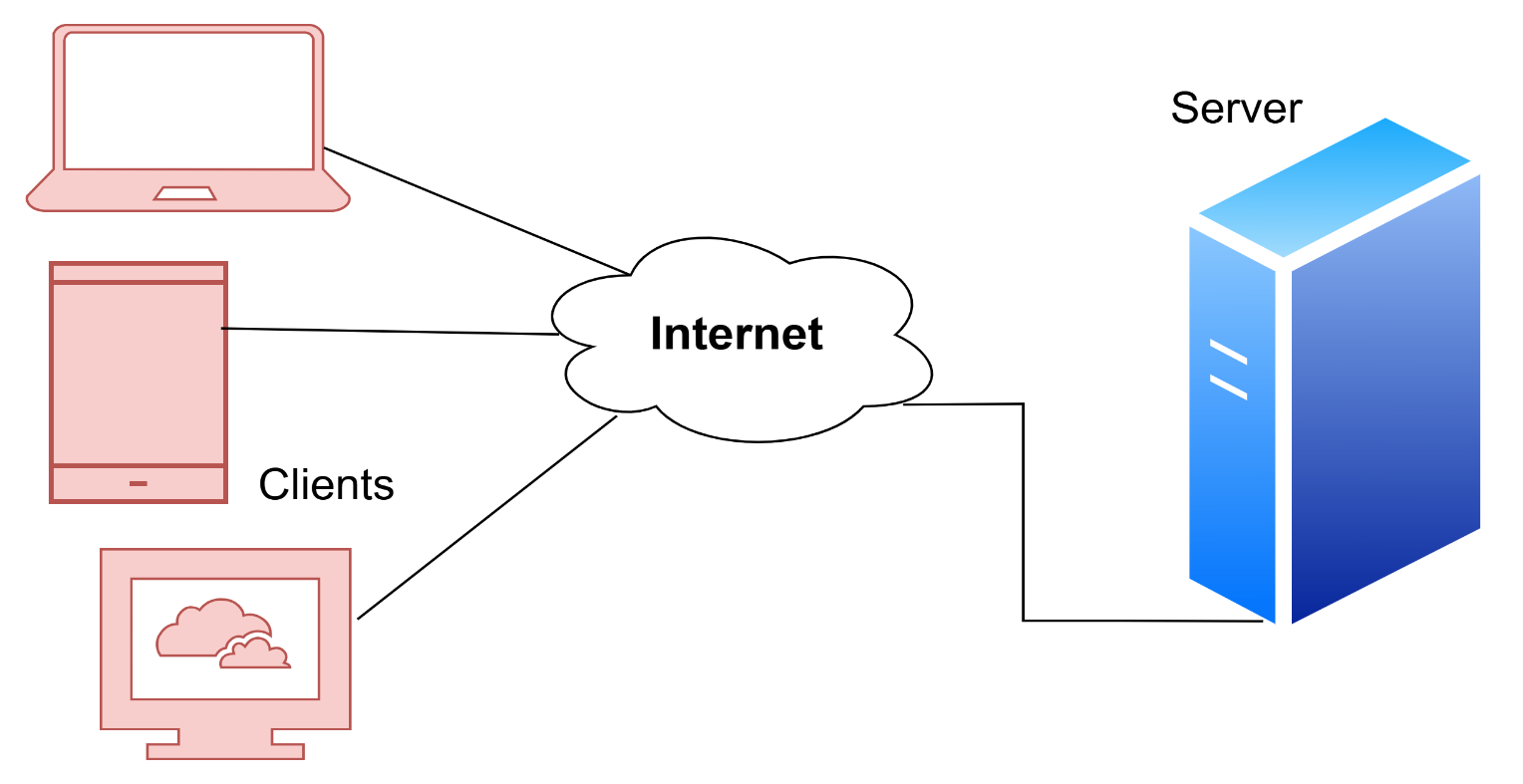
- The Server Receives your request and Processes it (e.g. find the webpages).

**5. Server sends response:**

- The server sends back the data over the internet using **TCP/IP** Protocols.

**6. Data Travels Back to Client:**

- The response travels Back through the network routers and your isp to your device.

**7. Browser Displays websites:** your browser receives the data and display the website your screen.****

**Que: Describe the roles of the client and server in web communication.?**

**Ans.** In web communication, the client (usually a web browser) initiates requests for information or services from the server. The server, in turn, processes these requests and sends back the requested information or performs the requested action.

**Client and Serve:**

**Que: Network Layers on Client and Server.?**

**Ans.** 1. Client:

* + Running and host.
  + Requests services.
  + E.g web browser

2. Server

* + Running and host
  + Provides services
  + E.g., web server

**Que: Explain the function of the TCP/IP model and its layers.?**

**Ans.** The **TCP/IP** mode (Transmission Control Protocol / Internet Protocol) is a set of rules that allows computers to communicate over the internet. It breaks down communication into layers to simplify how

Data is sent and received between devices.

. TCP/IP model is a four-layered architecture used to organize and manage network communication.

**1. Application layer:** The Application Layer is the top layer of the TCP/IP model. This I where users interact

with network services. It handles high – level protocols like email, Web browsing, and file transfer.

EX – HTTP, RTP, SMTP, DNS

**2. Transport Layer:** The Transport Layer is the second layer of TCP/IP model. Ensures reliable

Communication between devices. It breaks data into small packets and reassembles them at the

Destination.

Ex – TCP, UDP

**3. Internet Layer:** The Internet Layer is used for finding the best path for data to travel across different

networks so it can reach the right destination. It works like a traffic controller, helping data packets move from one network to another until they reach the correct device.

  The main job of this layer is routing deciding the best way for data to travel.

Ex – IP, ICMP;

**4. Network Access Layer:** The Network Access Layer is the bottom layer of the TCP/IP model. Manages a

Physical transmission of data over Network hardware likes cable and Wi-Fi.

Ex – Internet, Wi-Fi

**Que: Explain Client and Server Communication?**

**Ans.** Client-Server Communication means two computers talking to each other — one asks for something

(Client), and the other gives a response (Server).

**Client**

* A client is your device like a mobile phone, computer, or browser.
* It sends a request to the server.
* Example: When you open YouTube, your browser is the client.

**Server**

* A server is a powerful computer that stores websites, videos, data, etc.
* It receives the request from the client and sends back the right information.
* Example: YouTube’s server sends the video you want to watch

**Type Of Internet Connection**

1. **DSL** – Faster than dial-up, uses phone lines.
2. **Cable** – Uses TV cables, good speed.
3. **Fiber Optic** – Very fast, best quality.
4. **Wi-Fi** – Wireless internet through a router.
5. **Mobile Data (3G/4G/5G)** – Internet through SIM cards.
6. **Satellite** – Internet from satellites, used in remote areas.

**Que: Research different types of internet connections (e.g., broadband, fiber, satellite) and list their pros and cons?**

**Ans**. **1. DSL.**

Pros

* + widely available in urban and suburban areas.
  + This is affordable.

Cons.

* depends on distance Slower than cable and fiber .

**2. Cable Internet.**

Pros.

* + High speed
  + Widely available in cities and town
  + More stable than DSL

**3. Fiber – optic internet:**

Pros.

* extremely fast Symmetrical download and upload speed Very low latency and high reliability.

Con.

* limited available Higher cost compared to DSL/cable

**4. Satellite internet:**

Pros.

* Available almost anywhere
* Good for rural location

Cons.

* + Slower speed
  + High latency (delay)
  + Affected by weather

**5. Wireless internet:**

Pros.

* Easy to set up, no messy cable.
* You can use the internet anywhere withing the single range (home, office, public place).
* Multiple devices (laptop, phone) can connect at once.

Cons.

* Walls, distance, or other devices can weaken single strength.
* Wi – Fi signal Weaken after a certain distance.
* Usual not as fast or stable as Ethernet (wired) connection.

**6. Broadband over Power lines (BPL):**

Pros.

* No extra wiring needed.
* Can reach some areas not served by other methods.

Cons.

* Limited availability.
* Can suffer from interference and slower speeds.

**Que: How does broadband differ from fiber-optic internet?**

**Ans**. Broadband typically uses copper cables (DSL or coaxial) to deliver internet, while Fiber-optic uses glass or plastic Fibers that transmit data as light.  
Fiber-optic is much faster and more reliable than broadband.  
Broadband speeds can vary with distance and network traffic, but Fiber offers consistent high-speed.  
broadband is more widely available, especially in rural areas.  
Fiber is newer and often more expensive, but offers better performance.

**Protocols:**

* + A Network protocols is a group of rules accompanied by the network.
  + The protocol identifies the rules, syntax, semantics, and synchronization of communication and feasible error managing methods. In this article, we will discuss the different types of networking protocols.

**Type of Protocols:**

1. HTTP (Hyper Text Transfer Protocols) or HTTPS (HTTP Secure)

2. FTP (File Transfer Protocols)

3. Email Protocols (POP3, SMTP)

4. TCP (Transmission control Protocols)

5. UDP (User Datagram Protocols)

**Que: What are the differences between HTTP and HTTPS protocols?**

**Ans.** The main difference between HTTP and HTTPS is that HTTPS provides a secure, encrypted

Connection, while HTTP does not.

In HTTP URL start with “http://”. and HTTPS URL start with “https://”. HTTP faster than HTTPS.

**Application Security:**

1. Protects software applications from cyber threats.

2. Prevents common attacks like SQL Injection, XSS, etc.

3. Uses secure coding and regular security testing.

4. Tools like OWASP ZAP and Burp Suite help detect issues.

5. Ensures data protection, app reliability, and user trust.

**Que: Identify and explain three common application security vulnerabilities. Suggest possible solutions.?**

**Ans.**

**1. SQL Injection**

Explanation: Attackers insert malicious SQL code into input fields to access or manipulate the

database.

Solution:

* Use prepared statements or parameterized queries.
* Always validate and sanitize user inputs.

**2. Cross-Site Scripting (XSS)**

Explanation: Injecting malicious scripts into web pages, which run in users’ browsers.

Solution:

- Escape output properly (especially in HTML, JS, and URLs).

- Use Content Security Policy (CSP) to limit script execution.

**3. Broken Authentication**

Explanation: Poorly implemented login systems allow attackers to bypass authentication.

Solution:

- Use strong password policies and multi-factor authentication (MFA).

- Secure session management and avoid storing sensitive data in cookies.

**Que: What is the role of encryption in securing applications?**

**Ans.**

Role of Encryption in Securing Applications:

Encryption plays a crucial role in application security by protecting sensitive data from unauthorized access. It converts readable data (plaintext) into an unreadable format (ciphertext), ensuring that only authorized users with the correct decryption key can access the original information.

Key Roles:

1. Data Confidentiality – Prevents data leakage by keeping information secret.
2. Data Integrity – Helps ensure that data hasn’t been altered during transmission.
3. Authentication – Supports verifying the identity of users and systems.
4. Secure Communication – Encrypts data in transit (e.g., HTTPS, TLS).
5. Data Protection at Rest – Secures stored data (e.g., databases, files).

**Software Applications and Its Types**

* Software application is Preform by single or multiple tasks at the same period of time.

**Type of Software Applications:**

1. System Software
2. Application Software
3. Driver Software
4. Middleware Software
5. Programming Software

**1. System Software**

Purpose:  
- System software manages and controls computer hardware and provides a platform for running application software.

Key Features:

* Runs the computer and its hardware
* Works in the background
* Starts automatically when the system boots

Examples: - Notepad, Calculate etc.

**2. Application Software**

Purpose:  
- Application software helps users perform specific tasks or functions such as writing, editing, calculating, or designing.

Key Features:

* User-friendly interface
* Directly serves the user’s needs
* Needs system software to run

Examples: Microsoft Office, Paint, PowerPoint etc.

**3. Driver Software**

Purpose:  
- Driver software acts as a translator between the operating system and hardware devices. It enables the system to communicate with hardware correctly.

Key Features:

* + Controls and operates hardware
  + Essential for printers, keyboards, mouse, etc.
  + Installed automatically or manually

Examples: - Audio Driver, Video Driver etc.

**4. Middleware Software**

Purpose:  
- Middleware connects different applications or services so they can communicate or work together. It's used in distributed systems.

Key Features:

* Acts as a bridge between software components
* Used in enterprise and cloud environments
* Handles data management, messaging, authentication

Examples: - Database Middleware, application server Middleware etc.

**5. Programming Software**

Purpose:

Programming software is used by developers to write, test, and debug code for creating

applications and systems.

Key Features:

* Provides tools for coding and development
* Supports multiple programming languages
* Often includes compilers, interpreters, editors
* Examples: - Turbo c, Eclipse, Sublime etc.

**Que:** **Identify and classify 5 applications you use daily as either system software or application software.**?

**Ans.** - firstly I use Windows.

* I use code block for Programming.
* I use google for browsing the internet
* I use Microsoft word it helps to creating and edit documents
* I also use Whatsapp it let me send messages, calls etc

**Que: What is the difference between system software and application software?**

**Ans**. system software manages the computer's hardware and provides a platform for other software, while

application software allows users to perform specific tasks.

Low-level languages are used to write the system software.

High-level languages are used to write the application software.

**Software Architecture**

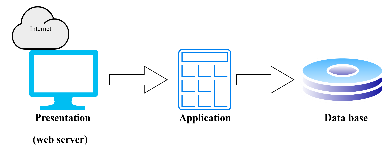
Software architecture is the blueprint of building software. It shows the overall structure of

the software.

This helps the software development team to clearly communicate how the software is going to be

built as per the requirements of customers.

**Que: Design a basic three-tier software architecture diagram for a web application.?**

****

**Que: What Is the significance of modularity in software architecture?**

**Ans**. Modularity in software architecture is significant because it enables breaking down complex systems

into smaller, manageable, and independent components (modules). This approach enhances various

aspects of software development and maintenance, including improved organization, reusability,

testing, scalability, and flexibility.

**Layers in Software Architecture**

**1. Presentation Layer:**

The presentation layer, also called the UI layer, handles the interactions that users have with the

software. It’s the most visible layer and defines the application’s layer overall look and presentation

to the end – user. This tier is the tier most accessible. Which anyone can use from their the client

devices. Like a desktop, laptop, mobile phone and tablet.

Focuses on the user interface and user experience.

**2. Application Layer:**

The application layer handles the main programs of the architecture. It includes the code definition

And most basic function of developed application.

This is layer that programmers spend most of their time in when working on the software. You can

use this layer to implement specific coordination logic that doesn’t align exactly with either the

presentation or business layer.

**3. Business Layer:**

The business layer, Also called the domain layer, is where are the application’s business logic operates.

Business logic a connection of rules that tell the system how to run an application. Based on the

Organization’s guidelines. This layer is essentially determines the behavior of the entire application.

After one action finishes. It tells the application what to do next.

**4. Persistence Layer:**

The persistence layer, also called the data access layer, acts as a protective layer.

It Contain the code that’s necessary to access the database layer. This layer also hold the set of codes

allow you manipulate various aspects of the database, such as connection details and SQL statement.

**5. Database Layer:**

The database layer is where the system stores all the data. It's the lowest tier in the software

architecture.

**Que:** **Create a case study on the functionality of the presentation, business logic, and data access layers of a given software system.**

Ans. 1. Presentation layer:

The presentation layer is responsible for the user interface (UI) of the e-commerce application.

2. Business Layer:

This layer contains the core business rules of the e-commerce application.

3. Data access layer:

This layer is responsible for interacting with the database to store and retrieve data.

**Que: Why are layers important in software architecture?**

ANS. Layers in software architecture separate the system into different parts based on functionality.

This improves clarity by isolating concerns like UI, logic, and data handling.  
 It makes the system easier to maintain and update over time.

Testing becomes simpler as each layer can be tested independently.

Reusability increases, especially for business logic and data functions.

Scalability improves since layers can scale separately.

Teams can work in parallel on different layers.

**Software Environments**

A software environment is the setup where software is developed, tested, and run.  
 It includes hardware, operating systems, tools, and libraries.  
**Que: Explore different types of software environments (development, testing, production). Set up a basic environment in a virtual machine.**

**Ans. 1. Development Environment:**

This is where developers write, debug, and test code.

**2**. **Testing Environment:**

This environment is used to conduct various types of testing, including functional, performance,

and security testing, to ensure the software meets quality standards before deployment.

**3. Production Environment:**

This is the live environment where the application is used by end-users. It requires careful

management and monitoring to ensure stability and availability.

Set up virtual basic machine Environment.

1.Install a Virtual Machine Platform

Download and install VirtualBox or VMware Workstation Player.

2.Download Ubuntu ISO

Grab Ubuntu Server/Desktop ISO from ubuntu.com.

3.Create a New Virtual Machine

Allocate memory 4 GB and disk space 20 GB.

Mount the Ubuntu ISO and begin installation.

**Que: Explain the importance of a development environment in software production**.?

**Ans.** A development environment is a place where a software developer can write, test, and debug code,

as well as access documentation and other resources that are needed to build the application.

**Source Code:**

**Que: Write and upload your first source code file to Git hub.?**

**Ans.** Python code:

Print (“my first git hub file!”)

Save this file in a folder my first program in python code.

1. Create a new GitHub repo that contains a README file.
2. Use Git to clone the GitHub repo locally.
3. Copy your project files into the folder created by the clone.
4. Perform a git add . and a git commit.
5. Chek git status.
6. Push your changes up to GitHub.

**Que: What is the difference between source code and machine code?**

**Ans:** Source code is human-readable instructions written in a programming language while machine code is

the low – level binary code that a computer processor directly executes.

**Git Hub and Introductions**

GitHub is a cloud-based platform used for version control and collaborative software development.  
It uses Git, a distributed version control system, to track changes in source code.  
Developers can store, share, and manage projects in repositories on GitHub.  
It allows multiple people to work on the same project using features like pull requests and branches.  
GitHub also supports project documentation, issue tracking, and code review tools.

**Que: Create a GitHub repository and document how to commit and push code changes.?**

**Ans.** Step 1. Create a GitHub Repository

1.Go to GitHub.com and log in.

2.Click the **+** icon in the upper-right corner select New repository.

3.Fill in

* Repository name.
* Public or Private visibility.
* Optional Add a README.

4.Click Create repository.

Step 2. Set git hub Locally

Navigate to your project directory in the terminal or command prompt and run: git init

Step 3. Add file to the Repository

Add your file:

gite add .

Step 4. Commit change:

Create a commit with a descriptive message explaining the changes made.

Code: git commit -m “ file name added “

Step 5. Push changes to git hub:

Upload your committed changes from the local repository to the remote GitHub repository.

Code: git push origin <branch>

**Que: Why is version control important in software development?**

**Ans. Version control is crucial in software development because it tracks changes to code over time,**

**allowing developers to collaborate effectively and manage different versions of their projects.**

**Student Account in Git hub**

**Que: Create a student account on Git hub and collaborate on a small project with a classmate.**

**Ans**. Step 1: Create a student git hub account

1.Go to git hub education.

2.Click get student benefits.

3.Sign in or create a github account.

4.Apply for the student developer pack by verifying your student status.

5.Once approved, you'll get free access to tools like git hub pro, canva, name cheap, and more.

Step 2: Start a small collaborative project

1.Create a git hub repository

* Sign in to git hub.
* Click new repository.
* Public or private visibility.
* Click create repository.

2. Add your Classmate as a collaborator

* Go to your repo settings collaborators.
* Type in your classmate’s git hub username.
* Click add collaborator they get an invite link.

3.Clone, commit & push code , Add code, commit, and push.

**Que: What are the benefit so using Git hub for students?**

**Ans**. GitHub helps students build coding skills by managing and sharing projects.

They get free access to premium tools via the GitHub Student Developer Pack.  
 It allows collaboration with classmates on real projects.  
 Students can showcase their work to future employers.  
 It teaches version control, a key industry skill.

**Type Of Software**

**Que: Create a list of software you use regularly and classify them into the following categories: system, application, and utility software.?**

**Ans.**  1. System software: Windows 11, android, macOS.

2. Application software: Google, Whatsapp, Spotify, MS word.

3. Utility software: Antivirus, backup tool.

**Que: What are the differences between open-source and proprietary software.?**

**Ans.** Differences between Open-Source and Proprietary Software:

* Open-source software has publicly available source code.
* Proprietary software keeps its source code private.
* Open-source is usually free to use.
* Proprietary often requires payment or license.
* Users can modify open-source software.
* Proprietary software cannot be legally modified.
* Open-source support is community-based.
* Proprietary offers official support from the vendor.

**GIT and GIT HUB training**

* Git is a tool to track and manage changes in code.
* GitHub is a platform to host Git repositories online.
* Use git clone, git add, git commit, and git push for basic operations.
* Create branches with git branch and merge them using git merge.
* GitHub enables team collaboration through pull requests and version control.

**Que: How does GIT improve collaboration in a software development team?**

**Ans. Git allows multiple developers to work on same project at the same time**

**It tracks changes, manages versions   and make easy to merge updates.**

Application Software

**Que: Write a report on the various types of application software and how they improve productivity.**

**Ans**. Word processor: Make creating/editing documents.

         Spreadsheet: Automate calculations and organize data.

        Graphic tools: Help design and edit media quickly.

        communication apps: Enable quick teamwork and sharing

**Que: What is the role of application software in businesses?**

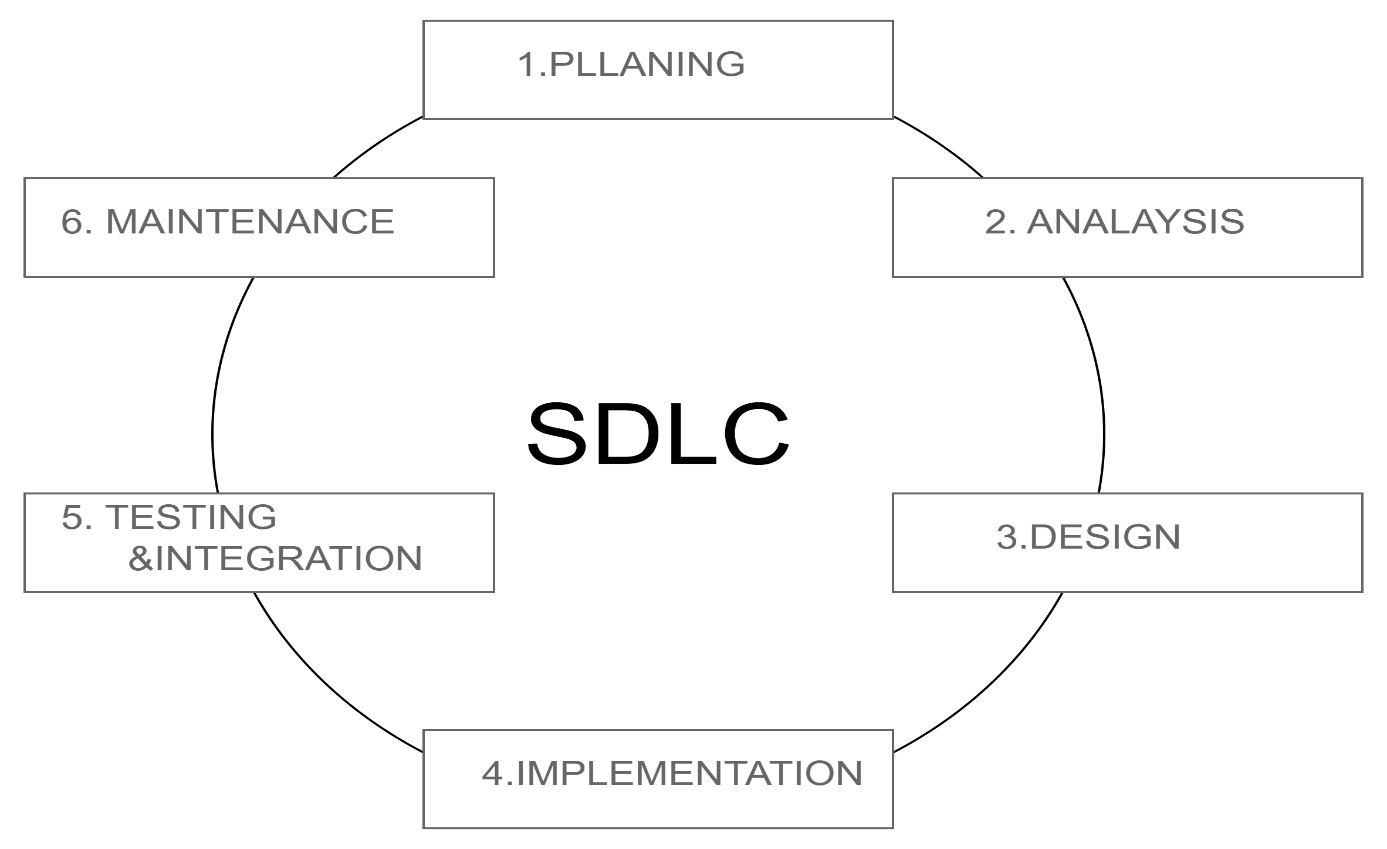
**Ans**. Application software automates tasks like billing and inventory.  
 It improves communication and collaboration within teams.  
 Data tools help businesses make better decisions.  
 Overall, it increases efficiency and productivity.

Enabling them to automate tasks, manage data, improve communication.

Software Development Process

**Que: Create a flow chart representing the Software Development Life Cycle (SDLC).?**

**Ans.**



**Que: What are the main stages of the software development process?**

**Ans.** These are the main stages of the software development.

1.Requirnment analysis

 2.Design

3.Development

4.Testing

5.Production

6.Maintanance

**Software Requirement**

**Que:** **Write a requirement specification for a simple library management system.**?

**Ans**. Ability to add and remove books from the library.

Ability to search for books in the library by title, author, or ISBN.

Ability to check out and return books.

Ability to display a list of all books in the library.

**Que: Why is the requirement analysis phase critical in software development?**

**Ans.** His phase defines what the software must do.

It ensures developers clearly understand what user needs.

Avoid mis understandings, reduces errors, saves time and cost later.

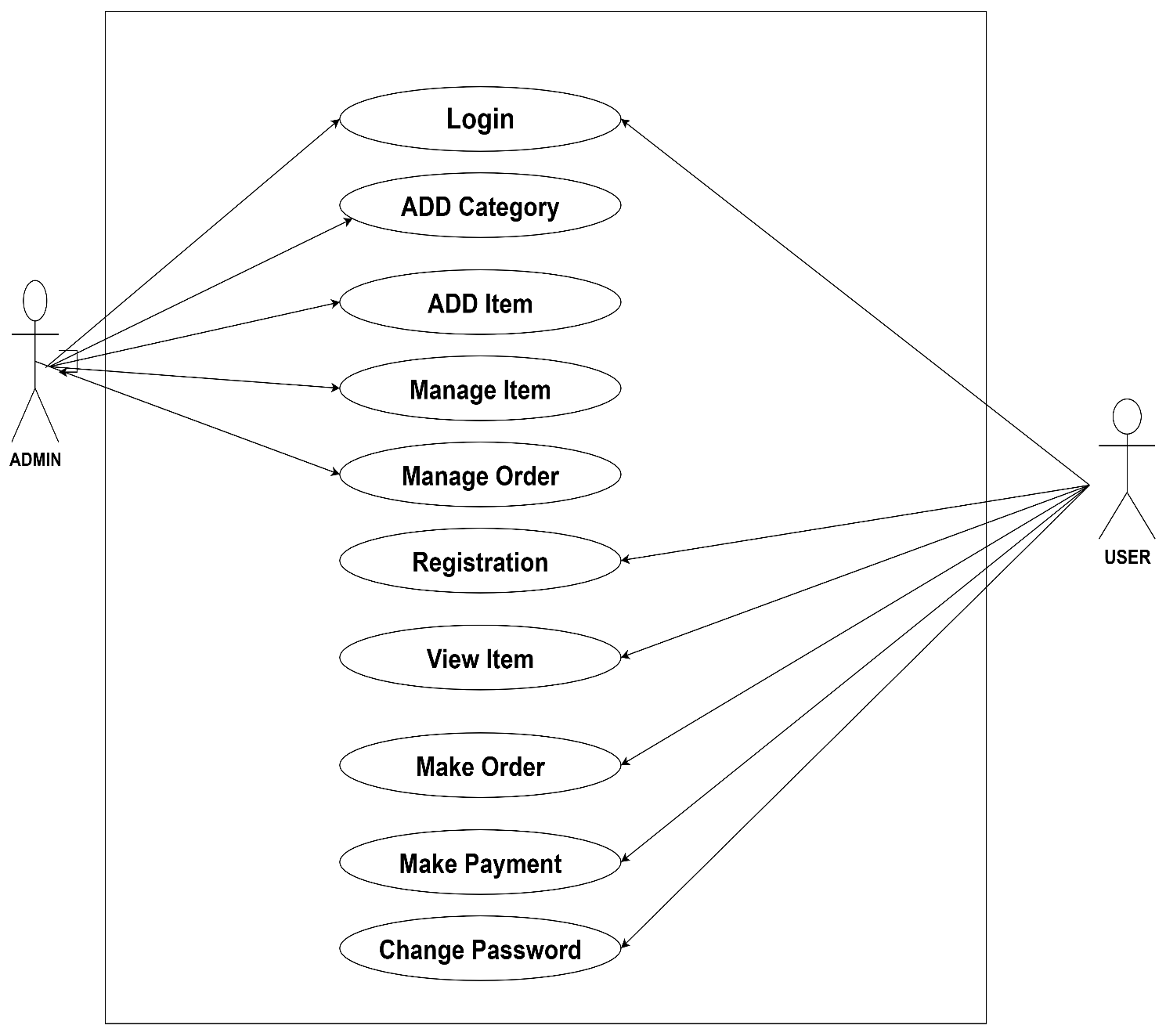
**Software Analysis**

**Que: What is the role of software analysis in the development process?**

**Ans. Software analysis helps understand user requirements and system needs.  
 It identifies functional and non-functional requirements.  
 It detects risks and constraints early in the process.  
 It provides a clear plan for design and development.  
 It ensures smooth communication among all stakeholders.**

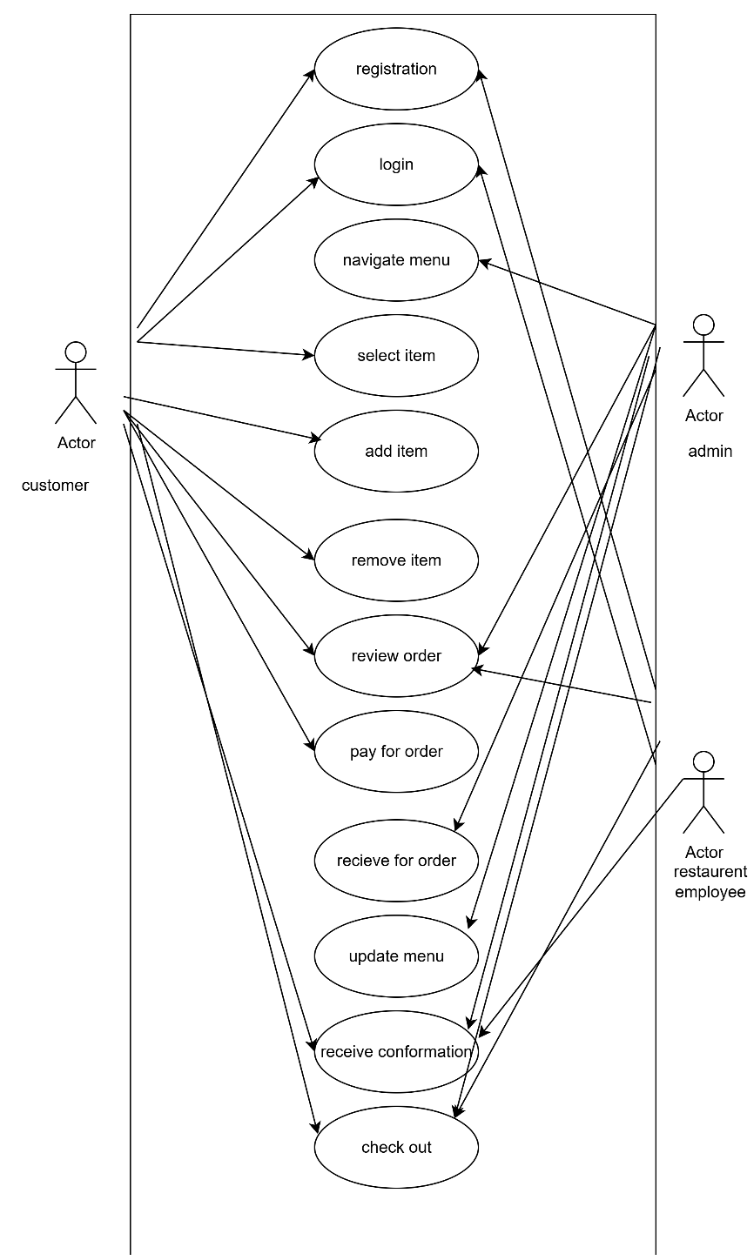
**Que: Performa functional analysis for an online shopping system.**

**Ans.**



**System Design**

**Que: Design a basic system architecture for a food delivery app?**



**Que: What are the key elements of system design?**

**Ans.** 1. Architecture: This defines the high-level structure of the system.

2.data flow: Understanding how data moves within the system.

3.scalability: This refers to the system's ability to handle increased load.

4.reliability: A reliable system consistently performs as expected under various conditions.

5.security:  Protecting the system from unauthorized access is paramount.

6.performance: Optimizing the system for speed and efficiency is essential.

7.maintainability: Designing a system that is easy to update, debug.

Software Testing

**Que: Why is software testing important?**

**Ans.** Software testing is important to ensure the software functions correctly and meets requirements.  
 It helps detect bugs early in the development process.  
 Testing improves the overall quality and reliability of the product.  
 It enhances user satisfaction by providing a better user experience.  
 Early testing reduces future maintenance and repair costs.

**Maintenance**

**Que: What types of software maintenance are there?**

**Ans.** 1. Corrective Maintenance – Fixes bugs and errors found after release.

2. Adaptive Maintenance – Updates software to work in new environments (e.g., OS or hardware

changes).

3. Perfective Maintenance – Enhances performance or adds new features based on user feedback.

4. Preventive Maintenance – Improves code structure or documentation to prevent future issues.

**Development**

**Que: What are the key differences between web and desktop applications?**

**Ans.** 1. Access: Web apps run in browsers and need internet; desktop apps run locally on a device.

2. Installation: Web apps don’t need installation; desktop apps must be installed.

3. Updates: Web apps update automatically; desktop apps require manual or system updates.

4. Platform Dependency: Web apps are usually cross-platform; desktop apps depend on OS.

5. Performance: Desktop apps often run faster due to local resources; web apps may be slower.

**Web Application**

**Que: What are the advantages of using web applications over desktop applications?**

**Ans.** Web applications offer several advantages over desktop applications.  
 They are accessible from any device with internet access, allowing flexibility and remote use.  
 No installation is needed, as they run directly in a web browser.  
 Updates are managed centrally, so users always have the latest version without manual action.  
 They are typically cross-platform, working on Windows, macOS, Linux, and mobile devices.  
 This makes maintenance easier and more cost-effective for developers and users alike.

**Designing**

**Que: What role does UI/UX design play in application development?**

**Ans. UI design** focuses on the look and feel of the app, including layout, colors, and buttons.  
 It ensures the interface is visually clear and easy to use.  
 **UX design** focuses on the overall user journey and experience.  
 It makes the app smooth, intuitive, and satisfying to use.

**Mobile Application**

**Que: What are the differences between native and hybrid mobile apps?**

**Ans.** Native apps are developed specifically for one platform like Android or iOS, using languages such as

Swift or Kotlin.

Hybrid apps are built using web technologies like HTML, CSS, and JavaScript and can run on

multiple platform.

Native apps offer better performance and a smoother user experience due to direct access to device

features.

Hybrid apps may be slightly slower as they run inside a web view.

Development of native apps is time-consuming and costly since separate codebases are required.

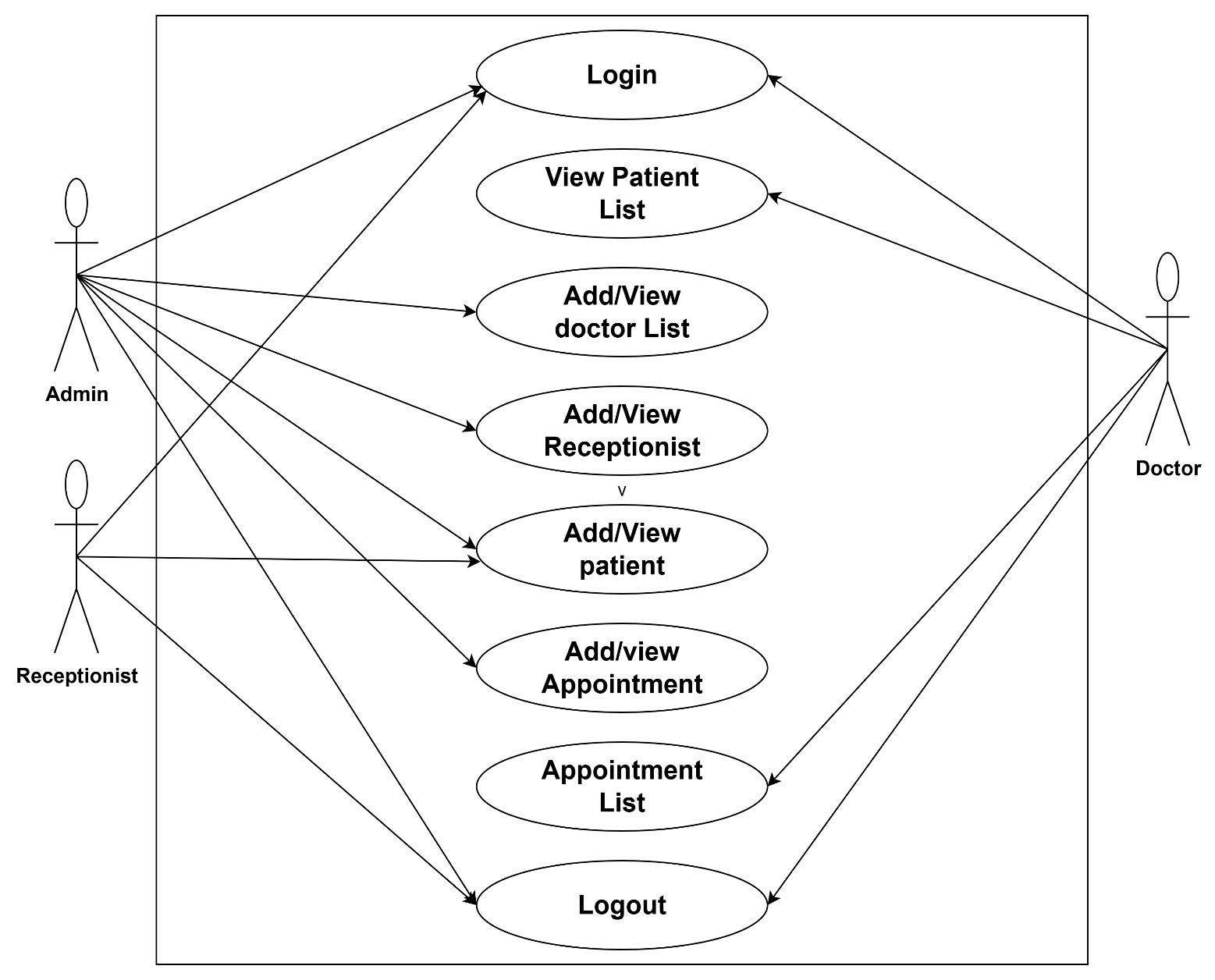
Hybrid apps are quicker and cheaper to build with a single codebase for all platforms.

Native apps follow platform-specific UI guidelines, enhancing user interaction.

Hybrid apps might not completely match native UI standards, affecting the user feel.

**DFD (Data Flow Diagram)**

**Que: Create a DFD for a hospital management system.**

****

**Que: What is the significance of DFDs in system analysis?**

**Ans**. Data Flow Diagrams (DFDs) visually represent the flow of data within a system.  
 They help in understanding system processes clearly and logically.  
 DFDs improve communication between developers and stakeholders.  
 They identify system inputs, outputs, and data storage needs.  
 DFDs assist in detecting inefficiencies and planning better system design.

**Desktop Application**

**Que: What are pros and cons of desktop applications compared to web applications?**

**Ans.** Pros of Desktop Applications:

1. Faster performance as they run locally on the system.
2. Can work offline without internet access.
3. Better access to hardware resources (e.g., file system, printers).

Cons of Desktop Applications:

1. Installation and updates must be done on each device.
2. Less accessible from multiple locations or devices.

**Flow Chart**

**Que: How do flow charts help in programming and system design?**

**Ans.**  Flowcharts help in programming and system design by providing a clear visual of the logical flow.  
 They simplify complex processes into understandable steps.  
 Flowcharts help identify errors or inefficiencies early in design.  
 They improve communication among team members and stakeholders.  
 Flowcharts also aid in debugging and maintaining the program later.