

[Module-1]

Theory Exercise

Q1) What is a Program?

A program is a set of written instructions that tells a computer how to perform specific tasks.

It is written in a programming language, translated into machine-understandable form by CPU.

Q2) Explain in your own words what a program is and how it functions What is programming?

Program is a Block of Code That tells a computer How to work.

Here is the working structure of program

Q3) What are the key steps involved in the programming process?

Types of programming language?

Steps involved in programming language is here below

{ Problem – Design – Implementation – Testing – Documentation – Deployment }

Types of programming language is

High level language - (Python , Java , c , c++)

Low level language – (Assembly and Machine level language)

Q4) What are the main differences between high-level and low-level programming languages?

Abstraction: High-level languages (e.g., Python, Java) offer strong abstraction from hardware and use human-friendly syntax; low-level languages (e.g., Assembly, machine code) map closely to CPU instructions.- **Portability:** High-level code is usually portable across platforms; low-level code is hardware-specific.

Productivity: High-level languages enable faster development and easier maintenance; low-level requires more effort but allows fine-grained control.

Performance & Control: Low-level languages can produce highly optimized, fast code and direct hardware control; high-level languages trade some performance for ease-of-use.

Use cases: High-level for application, web, and rapid development; low-level for device drivers, embedded systems, and performance-critical parts

Q5) Describe the roles of the client and server in web communication Network Layers on Client and Server ?

Client: The client (usually a web browser or app) initiates requests to access resources or services. It presents the user interface, sends HTTP requests, and displays or processes responses.

Server: The server listens for incoming requests, processes them (serving files, querying databases, running business logic), and returns responses (HTML, JSON, files). Servers manage resources, enforce security, and coordinate data persistence

Q6) Explain the function of the TCP/IP model and its layers Client and server ?

The TCP/IP model structures network communication into layers that separate concerns and standardize interactions:

Application layer: Interfaces for applications (HTTP, SMTP, DNS) — handles data formatting and high-level protocols.

Transport layer: Provides end-to-end communication and reliability (TCP for reliable ordered delivery, UDP for fast connectionless transmission).

Internet (Network) layer: Routes packets across networks (IP addressing and routing).

Link (Network Access) layer: Handles physical network hardware and local data transfer (Ethernet, Wi-Fi). Each layer adds/uses headers and passes data to adjacent layers so systems can interoperate across diverse networks

Client: A device or software that requests services (e.g., a web browser).

Server: A device or software that provides services (e.g., a web server).

Q7) Explain Client Server Communication Types of Internet Connections ?

Client–server communication is a model where clients request services and servers provide them over a network. The client initiates a request (e.g., HTTP GET), the server processes the request (may access storage, run logic), and the server sends back a response with status and data. Communication can be synchronous

(request/response) or asynchronous (push notifications, Web Sockets). Authentication, encryption, and session management are often used to secure and maintain state

Types of internet connections is :

- **Cable Internet**
- **Fiber-Optic Internet**
- **Satellite Internet**
- **Mobile Internet (3G, 4G, 5G)**
- **Fixed Wireless**
- **Wi-Fi**

Q8) How does broadband differ from Fiber-optic internet? Protocols

Broadband is a general term for high-speed internet access and includes technologies like DSL, cable, satellite, and Fiber. Broadband speeds and latency vary by technology.

Fiber-optic specifically uses glass/plastic Fibers to transmit data as light; it offers much higher bandwidth, lower latency, and greater reliability compared to DSL or cable. Fiber typically provides symmetrical upload/download speeds and better performance for heavy usage. Cable broadband often has higher contention (shared bandwidth) and asymmetric speeds, while DSL is slower and distance-sensitive. Satellite has high latency

Q9) What are the differences between HTTP and HTTPS protocols?

Application Security

Security: HTTP transmits data in plaintext; HTTPS encrypts data using TLS/SSL, preventing eavesdropping and tampering.

Port: Default HTTP uses port 80, HTTPS uses port 443.

Authentication/Integrity: HTTPS provides server (and optionally client) authentication via certificates and ensures message integrity.

Performance: HTTPS adds cryptographic overhead but modern TLS and HTTP/2 optimizations often mitigate performance costs.

Use cases: Sensitive data exchanges (logins, payments) must use HTTPS; plain HTTP is insecure for confidential information.

Q9) What is the role of encryption in securing applications?

Software Applications and Its Types

Encryption protects data confidentiality by converting readable data into ciphertext that only authorized parties can decrypt. In applications, encryption is used for:

Data in transit: TLS/HTTPS secures network communication from eavesdroppers.

Data at rest: Encrypting databases, files, or disks protects stored sensitive data if storage is compromised.

Authentication & Integrity: Cryptographic methods (digital signatures, HMACs) verify identity and detect tampering. Overall, encryption reduces risk of data exposure, supports compliance, and is a core part of defense-in-depth when combined with access controls, auditing, and secure key management

Q10) What is the difference between system software and application software? Software Architecture

System software manages and controls hardware and provides a platform for running applications (e.g., operating systems, device drivers).

Application software performs specific user-oriented tasks (e.g., word processors, web browsers). In short: system software enables the computer to run; application software helps users perform tasks

Q11) What is the significance of modularity in software architecture? Layers in Software Architecture

Modularity breaks a system into independent, well-defined components (modules). Benefits: easier development and testing, improved maintainability, parallel work by teams, reuse of modules, and simpler reasoning about system behavior. It also reduces complexity and localizes changes.

Layers in Software Architecture :

- Presentation
- Application
- Domain
- Data processing

Q12) Why are layers important in software architecture?

Layers separate concerns (e.g., presentation, business logic, data access), each with clear responsibilities. Benefits: improved organization, isolation of changes, easier testing, and clearer interfaces between parts. Layers enable swapping or upgrading one layer without affecting others.

Q13) Explain the importance of a development environment in software production?

A development environment provides the tools and configuration needed to write, build, test, and debug code (IDE, compilers, libraries, version control, test frameworks). It ensures developers reproduce a consistent setup, reduces 'it works on my machine' problems, and speeds up development through automation. Proper environments (dev, test, staging, production) support safe deployment and quality assurance

Q14) What is the difference between source code and machine code?

Source code: human-readable instructions written in a programming language (e.g., **Python, Java**).

Machine code: low-level binary instructions the CPU executes directly. Source code must be compiled or interpreted/translated into machine code before the processor can run it.

Q15) Why is version control important in software development?

Tracks history of changes, enabling rollback to earlier versions. Supports collaboration by merging work from multiple developers and resolving conflicts. Provides auditability, branching for feature development, and safer

releases. Facilitates code review, continuous integration, and reproducible builds.

Q16) What are the benefits of using Github for students?

Hosts repositories for free, making it easy to store and share projects. Enables collaboration, portfolio building, and showcasing work to employers. Integrates with CI/CD, issue tracking, and project management tools. Large community resources, tutorials, and open-source contributions to learn from.

Q17) What are the differences between open-source and proprietary software?

Open-source: source code is publicly available; users can inspect, modify, and redistribute. Encourages community contributions and transparency. Proprietary: source code is closed; usage and distribution restricted by the vendor. Trade-offs: open-source fosters flexibility and community support; proprietary often offers vendor support and warranties.

Q18) How does GIT improve collaboration in a software development team?

Allows multiple developers to work concurrently via branches and merges. Keeps a complete history of changes with authorship. Enables code reviews, automated testing on commits, and controlled integration of features. Reduces risk of accidental overwrites and makes integration and release processes more systematic

Q19) What is the role of application software in businesses?

Application software helps businesses perform specific tasks like accounting, data analysis, customer relationship management, and communication. It improves productivity, automates repetitive processes, reduces errors, and supports better decision-making.

Q20) What are the main stages of the software development process?

The main stages are:

- 1) Requirement analysis
- 2) System design
- 3) Implementation (coding)
- 4) Testing
- 5) Deployment
- 6) Maintenance.

Q21) Why is the requirement analysis phase critical in software development?

It ensures the development team understands the client's needs clearly. This phase defines project scope, avoids misunderstandings, reduces rework, saves time, and ensures the final product meets user expectations.

Q22) What is the role of software analysis in the development process?

Software analysis identifies the functional and non-functional requirements, constraints, and objectives. It helps create a

clear blueprint for the system, enabling accurate design and reducing the risk of errors.

Q23) What are the key elements of system design?

Key elements include: Architecture design (overall structure), Data design (database and data flow), Interface design (UI/UX), Component design (modules and sub-systems), and Security and performance considerations.

Q24) Why is software testing important?

Testing ensures the software is reliable, secure, and free of defects. It verifies that the product meets requirements, improves quality, prevents costly post-release fixes, and enhances user satisfaction.

Q25) What types of software maintenance are there?

Four main types:

- 1)** Corrective – fixing defects
- 2)** Adaptive – modifying software to work in a new environment
- 3)** Perfective – improving performance or adding features
- 4)** Preventive – making changes to prevent future problems.

Q26) What are the key differences between web and desktop applications?

Web applications run in browsers, require internet, and are platform-independent. Desktop applications run locally on a computer, may work offline, and are OS-specific.

Q27) What are the advantages of using web applications over desktop applications?

Accessible from anywhere, no installation needed, automatic updates, easier collaboration, and cross-platform compatibility.

Q28) What role does UI/UX design play in application development?

Key Roles of UI/UX Design in App Development:

1. Improves User Satisfaction

- A well-designed UI/UX makes the app intuitive and enjoyable, leading to higher user satisfaction.

2. Enhances Usability

- UX focuses on smooth navigation and easy interaction, helping users accomplish tasks quickly.

3. Boosts User Engagement

- Attractive UI and seamless UX keep users engaged and encourage repeat usage.

4. Reduces Development Costs

- Good design reduces the need for frequent changes and bug fixes by addressing user needs early.

5. Builds Brand Identity

- A consistent and visually pleasing UI strengthens brand recognition and trust.

Q29) What are the differences between native and hybrid mobile apps?

Platform:

Native – Built for one platform (iOS or Android)

Hybrid – Works on multiple platforms with one codebase

Language:

Native – Swift, Kotlin, Java

Hybrid – HTML, CSS, JavaScript (e.g., React Native, Ionic)

Performance:

Native – Faster and smoother

Hybrid – Slightly slower

Device Access:

Native – Full access to all device features

Hybrid – Limited access, may need plugins

Development Time:

Native – Longer (separate apps)

Hybrid – Shorter (single codebase)

Q30) What is the significance of DFDs in system analysis?

Data Flow Diagrams (DFDs) are visual models that show how data moves through a system — where it comes from, how it's processed, stored, and where it goes. Their significance includes:

- Clarify requirements: Make functional requirements explicit by showing processes, data stores, inputs and outputs.
- Simplify complexity: Decompose systems into levels for better focus.
- Improve communication: Easy for non-technical stakeholders to understand.
- Expose inefficiencies and errors: Highlight

redundant steps or missing processes early. - Basis for design and testing: Guide module, interface, and data store design. - Tool for documentation and maintenance: Serve as a persistent system blueprint

Q31) What are the pros and cons of desktop applications compared to webapplications ?

Pros of Desktop Applications: -

Performance: Faster and more responsive.

Offline availability: Can run without internet.

Rich access to system resources.

UX control: Optimized for the platform. Cons of Desktop Applications

Deployment & updates: Must be installed and updated manually.

Platform dependence: Requires separate builds for different OS.

Maintenance at scale: More complex. Pros of Web

Applications: Cross-platform reach. Centralized deployment and updates. - Easier scaling and collaboration Lower user friction.

Cons of Web Applications: - Dependence on network. Limited access to system resources.

Security & privacy exposure When to Choose: Desktop for high performance or offline work; Web for accessibility and easy updates.

Q32) How Flowcharts Help in Programming and System Design?

Flowcharts are graphical representations of algorithmic flow using standardized symbols.

They help by: - Making logic explicit: Show step-by-step **control flow.** - Reducing errors: Catch logical flaws early.

Improving communication: Easier for non-technical people to understand.

Aiding modular design: Clarify module interfaces.

Serving as documentation: Useful for maintenance.

Supporting testing and debugging: Helps design test cases and paths