

THEORY EXERCISE

1. Explain in your own words what a program is and how it functions.

- ☐ A program is basically a set of instructions written in a language that a computer can understand. To run a program, the computer follows these instructions one by one to perform tasks, like opening a file, calculating numbers, or displaying something on the screen.

2. What are the key steps involved in the programming process?

- ☐ The steps involved in the programming process are as follows:
 1. Understanding the problem – what needs to be solved.
 2. Planning – breaking the problem into smaller steps.
 3. Writing code – using a programming language.
 4. Testing & Debugging – finding and fixing errors.
 5. Execution – running the program to get results.
 6. Maintenance – updating or improving later.

3. What are the main differences between high-level and low-level programming languages?

- ☐ High-level languages are closer to human language and easier to write and understand, such as Python or Java. They handle a lot of complex stuff for you behind the scenes.
- ☐ Low-level languages, like Assembly or machine code, talk directly to the hardware and are harder to write but faster, and give you more control over the computer.

4. Describe the roles of the client and server in web communication. Network Layers on Client and Server.

- ☐ A client is a program that runs on the local machine, requesting service from the server
- ☐ A server is a program that runs on a remote machine, providing services to clients.
- ☐ Network layers on a client and server are typically described using the OSI model or the TCP/IP model. The OSI model has seven layers: Physical, Data Link, Network, Transport, Session, Presentation, and Application. The TCP/IP model has four layers: Network Access, Internet, Transport, and Application. These layers work together to facilitate communication between devices.

5. Explain the function of the TCP/IP model and its layers. Client and Servers?

- ☐ The TCP/IP model helps computers communicate over the Internet.
 - Application Layer – services like HTTP, FTP, Email.
 - Transport Layer – ensures reliable delivery (TCP) or fast delivery (UDP).
 - Internet Layer – uses IP addresses to send data.
 - Network Access Layer – handles physical connection (Ethernet, Wi-Fi).

- Client and server both use these layers to send and receive messages.

6. Explain Client-Server Communication.

- When you type a website address, your client (browser) sends a request to the server. The server processes it (for example, fetches the web page) and sends the result back to the client.

7. How does broadband differ from Fiber-optic internet? Protocols.

- Broadband and Fiber optic internet are related but distinct concepts. Broadband is a general term for high-speed internet access, while Fiber optic internet is a specific technology that uses Fiber optic cables to deliver that high-speed access.

8. What are the differences between HTTP and HTTPS protocols?

- The primary difference between HTTP and HTTPS is security
 - HTTP (Hypertext Transfer Protocol): Transfers web data but is not secure.
 - HTTPS (Hypertext Transfer Protocol Secure): Adds encryption (SSL/TLS), making data safe from hackers.
- Think of HTTP as sending a postcard, and HTTPS as sending a sealed envelope.

9. What is the role of encryption in securing applications?

- Encryption scrambles the data into an unreadable form so that only the authorized person can read it. It protects sensitive information like passwords, payments, and personal data from hackers.

10. What is the difference between system software and application software?

- System software manages the computer's hardware and provides a platform for other software, while application software is designed to perform specific tasks for the user.

11. What is the significance of modularity in software architecture?

- Modularity means breaking software into smaller, independent parts (modules). This makes development easier, allows reusing code, simplifies debugging, and improves teamwork.

12. Why are layers important in software architecture?

- Layers keep things organized by separating functions into different levels. This separation helps developers focus on one thing at a time and makes the software more flexible and easier to fix.
- If one layer changes, others are less affected.

13. Explain the importance of a development environment in software production.

- ☐ A development environment provides tools to write, test, and debug programs. A development environment is where programmers write, test, and debug their code.
- ☐ It increases productivity and reduces errors by giving features like auto-suggestions, debugging, and version control.
- ☐ Ex. VS Code, Eclipse, or PyCharm)

14. What is the difference between source code and machine code?

- ☐ Source code is the human-readable instructions programmers write in languages
- ☐ Ex. Python or Java.
- ☐ Machine code is the super low-level instructions made of 0s and 1s that the computer's processor actually understands and executes.
- ☐ Compilers or interpreters convert source code into machine code.

15. Why is version control important in software development?

- ☐ Version control keeps track of changes in code, allows teamwork without overwriting each other's work, and helps roll back to previous versions if something goes wrong.

16. What are the benefits of using GitHub for students?

- ☐ GitHub is awesome for students because it's like a social network for coding. It helps you store your projects online, track changes, and collaborate with others easily. Students can benefit from GitHub by gaining practical experience with version control and essential skills for future careers in tech.

17. What are the differences between open-source and proprietary software?

- ☐ Open-source software is like a community project — anyone can see, use, or change the code.
- ☐ Proprietary software is owned by a company, and the code is kept secret.
- ☐ You usually have to pay for proprietary software, while open-source is often free.

18. How does GIT improve collaboration in a software development team?

- ☐ Git allows multiple developers to work on the same project without messing things up. They can create branches, merge changes, and track history, which makes teamwork smooth and organized.
- ☐ This makes teamwork smoother and saves a lot of headaches.

19. What is the role of application software in businesses?

- ☐ Application software helps businesses in operations like accounting, sales, customer management, data analysis, and communication, increasing efficiency and productivity.

20. What are the main stages of the software development process?

- ☐ Planning: Defining the project scope, requirements, and objectives

- ☐ Design: Creating the architecture, user interface, and technical specifications.
- ☐ Coding: Writing the code and building the software.
- ☐ Testing: Evaluating the software for functionality, performance, and security.
- ☐ Deployment: Releasing the software to users.
- ☐ Maintenance: Providing ongoing support, bug fixes, and updates

21. Why is the requirement analysis phase critical in software development?

- ☐ This phase sets the foundation. If the requirements are unclear, the final software will not solve the real problem. Clear requirements save cost, time, and rework.

22. What is the role of software analysis in the development process?

- ☐ Software analysis breaks down the problem and figures out how the software should behave. It helps decide what features to build, what technology to use, and how to structure the project.

23. What are the key elements of system design?

- ☐ Architecture (overall structure).
- ☐ Database design.
- ☐ User Interface design.
- ☐ Security.
- ☐ Performance and scalability planning.

24. Why is software testing important?

- ☐ Testing ensures the software is error-free, reliable, and meets user needs. It prevents failures after launch and builds trust in the application.

25. What types of software maintenance are there?

- ☐ There are mainly three types:
 1. corrective (fixing bugs).
 2. adaptive (updating software for new environments or systems).
 3. perfective (improving performance or adding features).
- ☐ Maintenance keeps software useful and running smoothly over time.

26. What are the key differences between web and desktop applications?

☐ **Key Differences between Web and Desktop Applications**

Installation:

Web App → Runs in browser, no installation needed.

Desktop App → Must be installed on a computer.

Accessibility:

Web App → Accessible from anywhere with internet.

Desktop App → Limited to the device where installed.

Updates:

Web App → Auto-updated by server.

Desktop App → User must update manually.

Performance:

Web App → Depends on internet speed.

Desktop App → Usually faster, runs offline.

Storage:

Web App → Stores data online (cloud).

Desktop App → Stores data on local device.

27. What are the advantages of using web applications over desktop applications?

- ☐ Web: No need to install on every machine.
- ☐ Accessible from anywhere with internet.
- ☐ Easy updates (done on server, not user devices).
- ☐ Works across platforms (Windows, Mac, Mobile).

28. What role does UI/UX design play in application development?

- ☐ UI (User Interface) and UX (User Experience) design make sure the app is easy and pleasant to use. A well-designed app improves user satisfaction and increases adoption.

29. What are the differences between native and hybrid mobile apps?

- ☐ Native apps are built specifically for one platform and usually perform better and have access to device features.
Ex. like iOS or Android
- ☐ Hybrid apps use web technologies, allowing them to run on multiple platforms with a single codebase, but might not be as fast or smooth run on multiple platforms, and are cheaper but sometimes slower.
Ex. HTML, CSS, JS

30. What is the significance of DFDs in system analysis?

- ☐ Data Flow Diagrams (DFDs) show how data moves through a system. It helps understand the process flow, inputs, and outputs clearly before coding starts, making it easier to spot problems or plan improvements.

31. What are the pros and cons of desktop applications compared to web applications?

- ☐ Pros of desktop apps: Faster, can work offline, more powerful features.
Cons: Need installation, updates are harder, and platform-dependent.
- ☐ Pros of web apps: Easy to access anywhere, no installation, easy updates.
Cons: Needs internet, sometimes slower.

32. How do flowcharts help in programming and system design?

- ☐ Flowcharts visually represent the steps of a program or system using symbols. They make logic easier to understand, debug, and explain to others before writing actual code.