

Name: Paramprakash Makwana

Module 1 – Overview of IT Industry (Theory)

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

Ans:- A program is a set of instructions written to tell a computer how to perform a specific task. It functions by the computer reading these instructions, processing data accordingly, and producing the desired output.

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

Ans:- The key steps in the programming process are:

1. Understanding the Problem – Analyze what the program needs to do.
2. Designing a Solution – Plan the logic or algorithm for solving the problem.
3. Writing Code – Implement the solution in a programming language.
4. Testing and Debugging – Check for errors and fix them to ensure the program works correctly.
5. Maintenance – Update and improve the program over time as needed.

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

Ans:- High-level languages are easier to read, write, and understand, and are closer to human language. Low-level languages are closer to machine language, offer more control over hardware, but are harder to write and understand.

1.4) Describe the roles of the client and server in web communication.

Ans:- The client is the device or application that requests data or services, such as a web browser. The server receives these requests, processes them, and sends back the required response, like a web page or data.

1.5) Explain the function of the TCP/IP model and its layers.

Ans:- The TCP/IP model defines how data is transmitted over a network and ensures devices can communicate reliably. Its layers are:

1. Application Layer – Provides network services to applications, like web browsing (HTTP) or email (SMTP).
2. Transport Layer – Ensures reliable data delivery using protocols like TCP, or faster, connectionless delivery with UDP.
3. Internet Layer – Handles addressing and routing of data packets using IP.
4. Network Access Layer – Manages the physical transmission of data over the network, including hardware and protocols for local connections.

1.6) Explain Client Server Communication

Ans:- Client-server communication works on a request-response model. The client sends a request to the server, the server processes it, and then sends a response back to the client.

1.7) How does broadband differ from fiber-optic internet?

Ans:- Broadband is a general term for high speed internet delivered over copper cables, cable TV lines, or wireless networks. Fiber optic internet uses fiber cables to transmit data as light, making it much faster, more reliable, and better for heavy usage.

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

Ans:- HTTP and HTTPS are used to transfer data between a browser and a server, but the key difference is security.

HTTP sends data in plain text, so it can be intercepted or altered. HTTPS encrypts the data using SSL or TLS and verifies the website's identity, making it much safer for sensitive information like passwords and payments.

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

Ans:- Encryption protects application data by converting it into an unreadable form so only authorized users can access it. It helps keep sensitive information safe during storage and transmission, even if the data is intercepted.

1.10) What is the difference between system software and application software?

Ans:- System software manages and controls the computer's hardware and provides a platform for other programs to run, like an operating system.

Application software is designed to help users perform specific tasks, such as word processing, browsing the web, or managing data.

1.11) What is the significance of modularity in software architecture?

Ans:- Modularity divides software into independent, self-contained modules, each handling a specific task. This makes the system easier to develop, test, maintain, and scale, and allows changes in one module without affecting others.

1.12) Why are layers important in software architecture?

Ans:- Layers are important in software architecture because they separate different responsibilities within a system. This makes the software easier to develop, maintain, test, and scale, since changes in one layer usually do not impact the others.

1.13) Explain the importance of a development environment in software production.

Ans:- A development environment provides the tools and setup needed to write, test, and debug software efficiently. It helps developers work consistently, catch errors early, and build reliable software before it is released.

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

Ans:- Source code is the human readable code written by a programmer using a programming language.

Machine code is the low level binary code made of 0s and 1s that the computer's processor can directly execute.

1.15) Why is version control important in software development?

Ans:- Version control is important because it tracks changes to code over time and allows developers to work together safely. It helps prevent data loss, makes it easy to fix mistakes, and lets teams return to earlier versions when needed.

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

Ans:- GitHub helps students manage and track their code using version control. It makes collaboration easy, provides a portfolio to showcase projects, and helps students learn real world development practices used in the industry.

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

Ans:- Open source software allows users to view, modify, and share its source code freely. Proprietary software keeps its source code private and restricts how users can use, modify, or distribute it.

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

Ans:- Git improves collaboration by allowing multiple developers to work on the same project at the same time without overwriting each other's work. It tracks every change, makes it easy to merge contributions, and helps resolve conflicts in a controlled way.

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

Ans:- Application software helps businesses perform specific tasks efficiently, such as managing finances, communicating with clients, analyzing data, or handling inventory. It increases productivity, reduces errors, and supports decision-making.

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

Ans:- The main stages of the software development process are:

1. Requirement Analysis – Understanding what the users need.
2. Design – Planning the software structure and components.
3. Implementation (Coding) – Writing the actual code.
4. Testing – Checking for errors and ensuring the software works as intended.
5. Deployment – Releasing the software for use.
6. Maintenance – Updating and fixing the software over time.

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

Ans:- Requirement analysis is critical because it defines exactly what the software must do. Clear requirements prevent misunderstandings, reduce errors, and ensure the final product meets user needs and expectations.

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

Ans:- Software analysis involves examining user needs, system requirements, and existing processes to define what the software should achieve. It ensures the development team understands the problem clearly, which helps design an effective and reliable solution.

1.23) What are the key elements of system design?

Ans:- The key elements of system design are:

1. Architecture Design – Defines the overall structure and components of the system.
2. Data Design – Plans how data will be stored, organized, and accessed.
3. Interface Design – Specifies how users and other systems will interact with the software.
4. Component Design – Details the functionality of individual modules or components.
5. Security and Performance Considerations – Ensures the system is safe, efficient, and scalable.

1.24) Why is software testing important?

Ans:- Software testing is important because it identifies errors and defects before the software is released. It ensures the software works correctly, meets requirements, and provides a reliable and secure experience for users.

1.25) What types of software maintenance are there?

Ans:- There are four main types of software maintenance:

1. Corrective Maintenance – Fixing bugs or errors discovered after release.
2. Adaptive Maintenance – Updating software to work with new hardware, operating systems, or environments.
3. Perfective Maintenance – Improving performance, usability, or adding new features.
4. Preventive Maintenance – Making changes to prevent future problems or improve reliability.

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

Ans:- Web applications run in a web browser and can be accessed from any device with internet access. They are easier to update and maintain but depend on a network connection.

Desktop applications are installed on a specific computer and run locally. They can work offline and often offer faster performance but require separate installations and updates on each device.

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

Ans:- Web applications offer easy access from any device with a browser, require no installation, and are easier to update and maintain. They also support collaboration and real-time data sharing across multiple users.

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

Ans:- UI/UX design ensures that an application is user-friendly, intuitive, and visually appealing. Good design improves user satisfaction, reduces errors, and makes the software easier and more enjoyable to use.

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

Ans:- Native apps are built for a specific platform (like iOS or Android) using platform-specific languages, offering better performance and access to device features.

Hybrid apps are built using web technologies (HTML, CSS, JavaScript) and run on multiple platforms through a single codebase. They are easier to develop and maintain but may have lower performance and limited access to device features.

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

Ans:- Data Flow Diagrams (DFDs) are significant because they visually represent how data moves through a system. They help analysts understand processes, identify inefficiencies, and communicate system requirements clearly to stakeholders.

1.31) What are the pros and cons of desktop applications compared to web applications?

Ans:- Desktop Applications

Pros:

- Can work offline without an internet connection.
- Often faster and more responsive.
- Greater access to hardware features.

Cons:

- Must be installed on each device separately.
- Updates require manual installation on each device.
- Less convenient for collaboration across multiple users.

Web Applications

Pros:

- Accessible from any device with a browser.
- Easy to update and maintain centrally.
- Supports real-time collaboration.

Cons:

- Requires an internet connection.
- Performance can be slower than desktop apps.
- Limited access to some device hardware features.

1.32) How do flowcharts help in programming and system design?

Ans:- Flowcharts help by visually representing the steps and decision points in a process or program. They make it easier to understand, plan, and communicate the logic, identify errors early, and organize complex systems before coding.