**Module 1 – Overview of IT industry**

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

Ans: A program is computer understandable language which executes a set of instructions to perform a specific task.

**Functions:** 1. Written in programming language

2. Translation in machine code.

3. Execution by computer

🡪Input: User writes input

🡪Processing: System performs calculations or make decision

🡪Output: Output is shown to user.

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

Ans: Programming process involves following steps:

A: Define a problem: Firstly, understand what program is meant to define for. This involves gathering information, clarifying objectives.

B: Plan the solution: Develop the logical plan for designing the solution. By using techniques like pseudocode, flowchart etc.

C. Code the program: Translate the designed solution into a specific programming language.

D: Test the program: Run the program with various inputs to identify and fix errors and ensure it produces the correct outputs in all situations. Testing includes both functional testing and performance testing.

E: Document the program: Create a simple and small documentation of program giving explanation about the problem and its solution.

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

Ans: High level language:

Programmer friendly language, easy to understand, simple to debug, simple to maintain, Less memory efficient

Low level language:

Machine friendly language, tough to understand, complex to maintain comparatively, complex to debug comparatively, high memory efficient.

4. Describe the roles of the client and server in web communication.

Ans: In web communication, the client and server have distinct roles. The client, typically a web browser, initiates requests for resources like web pages or data from the server. The server, in turn, receives these requests, processes them, and sends back the requested information or data. This interaction follows a "request and response" pattern.

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

Ans: TCP/IP has 4 layers:

1: Application layer: This layer provides network services directly to the end-user applications. It facilitates communication between software applications and the underlying network.

2: Transport layer: This layer is responsible for end-to-end communication, error recovery, and flow control. It ensures that data is delivered reliably and in the correct order.

3: Internet layer: This layer handles the routing of data packets across the network. It defines the addressing and routing methods used to send data from the source to the destination.

4: Network Access layer: This layer is responsible for the physical transmission of data over the network medium. It deals with the hardware addressing and the protocols that operate on the local network.

6: Explain Client Server Communication.

Ans: Client-server communication is a fundamental computing model where a client, such as a user's computer or application, requests services or resources from a server, which then provides those services. This interaction is characterized by a request-response pattern, with the client initiating the communication and the server responding.

Process:

Request: Client or user request to server.

Process: Server receives the request and processes it. This may involve querying a database, performing calculations or file accessing.

Response: After processing the server sends response back to client.

Display: The client receives the response and takes appropriate action, such as

displaying the data to the user or updating the user interface.

7: How does broadband differ from fiber-optic internet?

Ans: Broadband is a general term for high-speed internet access, while fiber-optic internet is a specific type of broadband technology that uses fiber-optic cables to transmit data as light signals. Fiber-optic internet is generally faster and more reliable than other broadband options like DSL cable, which use copper wires.

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

Ans:

HTTP: Hypertext transfer protocol, Data is transmitted in plain text, making it vulnerable to interception and attacks (e.g., man-in-the-middle attacks), port 80, begin with "http://", Does not ensure data integrity; data can be altered during transmission.

HTTPS: Hypertext transfer protocol secure, Data is encrypted using SSL/TLS (Secure Sockets Layer/Transport Layer Security), providing a secure channel for communication and protecting data from eavesdropping, port 443, begin with "https://", Ensures data integrity, meaning that data cannot be modified or corrupted during transfer without detection.

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

Ans: Encryption is a staple of modern IT paradigms. It provides an additional layer of security and ensures that even if an attacker gains access to the encrypted data, it remains inaccessible and meaningless. It transforms readable data into an unreadable format, ensuring that even if intercepted, the information remains confidential and unusable without the decryption key.

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

Ans:

System Software:

🡪System software is designed to manage and control computer hardware and provide a platform for running application software.

🡪System software interacts directly with the hardware components.

🡪While users may interact with system software, its primary function is not to perform tasks for the user but to manage system resources.

🡪Eg: OS, Device drivers, Utility programs

Application software:

🡪Application software is designed to help users perform specific tasks or applications.

🡪Application software relies on system software to access hardware resources.

🡪Application software is designed for direct user interaction, allowing users to perform specific tasks, create content, or manage information.

🡪Eg: Microsoft word, Excel, Games etc.

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

Ans: Modularity in software architecture is significant because it breaks down complex systems into smaller, manageable, and independent components called modules. This approach simplifies development, testing, and maintenance, enhancing code organization, reusability, and scalability. By isolating changes to specific modules, it also improves fault tolerance and allows for easier updates and modifications.

12: Why are layers important in software architecture?

Ans: Layers in software architecture are crucial because they promote component based segment, reusability, and maintainability. By dividing a system into distinct layers, each with specific responsibilities, developers can isolate changes, making it easier to manage, update, and test different parts of the application independently. This leads to more efficient development, reduced costs, and a more robust and adaptable software system.

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

Ans: A development environment is crucial in software production as it provides a dedicated space for developers to write, test, and refine code without impacting live systems or users.

It allows for experimentation, error correction, and feature development in a safe and isolated environment before code is deployed to production.

This isolation helps prevent bugs from reaching users and ensures a smoother, more efficient development process.

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

Ans:

Source code:

🡪Set of instructions written in human readable format. Eg (Python, C, etc)

🡪Uses syntax and semantics of respective programming language.

🡪Primary purpose is to define logic.

🡪It is generally portable across different platforms, as long as the target platform supports the programming language used.

Machine code:

🡪Low level code directly executed by computer CPU. Eg (0s, 1s)

🡪Instruction corresponds to specific operations that the CPU can perform, making it difficult for humans to interpret.

🡪The purpose of machine code is to be executed by the computer's hardware.

🡪It is platform-specific, meaning that machine code generated for one type of CPU architecture cannot be executed on another without recompilation or translation.

15: Why is version control important in software development?

Ans: Version control is essential in software development as it enables collaboration among developers, allowing them to work simultaneously without conflicts. It maintains a history of changes for tracking project evolution and acts as a backup system for quick recovery of previous versions. By supporting branching, it allows experimentation without affecting the main codebase and enhances code quality through reviews and integration with continuous integration tools. Version control also helps manage conflicts, provides documentation through commit messages, and integrates with other tools for automated workflows, ultimately improving efficiency and adaptability in development.

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

Ans: The GitHub offers numerous benefits for students, including free access to valuable tools, collaboration opportunities, and a platform to build a professional portfolio. It's a gateway to the professional world, providing resources and exposure to help students grow, whether they're beginners or advanced developers.

Here are some benefits listed:

🡪Portfolio development, Collaboration, Open source contribution, Learning resources etc.

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

Ans:

Open-Source Software:

🡪Source code is available for public view, modification and redistribution.

🡪Often free to use but may involve cost for support or specific versions.

🡪Highly customizable due to access to source code.

🡪Relies on community contributions for development, bug fixes, and support.

🡪Eg: Linux, Android etc

Proprietary software:

🡪Source code is closed and not publicly accessible.

🡪Generally, involves licensing fees.

🡪Limited customization options.

🡪Support is primarily provided by vendors.

🡪MS office, Adobe photoshop.

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

Ans: Git enhances collaboration in software development teams by allowing multiple developers to work on the same codebase simultaneously, track changes, and merge contributions seamlessly. Its distributed nature supports efficient workflows, enabling teams to manage complex projects and maintain a clear project history.

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

Ans: Application software is essential for businesses as it automates tasks, manages data, and enhances communication. It improves customer relationships through CRM systems, supports financial management, and aids in project and HR management. Additionally, it facilitates marketing efforts and provides analytics for informed decision-making, ultimately driving efficiency and growth.

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

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

**🡪Planning**: Define project goals, scope, and requirements.

**🡪Analysis**: Gather and analyze requirements from stakeholders.

**🡪Design**: Create architecture and design specifications.

**🡪Implementation**: Write and test the code.

**🡪Testing**: Validate the software for defects and ensure it meets requirements.

**🡪Deployment**: Release the software to users.

**🡪Maintenance**: Provide ongoing support and updates as needed.

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

Ans: Requirement analysis is critical because here we define the needs of users and stakeholders. It helps to identify and clarify the requirement, reduces risk management, minimizes the costly changes later on.

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

Ans: Software analysis, also known as requirements analysis, is a crucial phase in software development that involves understanding, defining, and documenting the requirements for a software system before development begins. It essentially acts as a bridge between the business needs and the technical implementation, ensuring that the final product meets user expectations and project goals.

23: What are the key elements of system design?

Ans: Key element in system design includes security, performance, integration, data design, architecture etc.

24: Why is software testing important?

Ans: Software testing is crucial because it ensures a product's quality, reliability, and security before release. By identifying and fixing bugs early, testing helps reduce costs associated with later-stage fixes and potential failures. It also enhances user satisfaction, protects against security vulnerabilities, and can prevent costly mistakes.

25: What types of software maintenance are there?

Ans: There are four main types of software maintenance:

**🡪Corrective Maintenance**: Fixing defects or bugs found in the software after deployment.

**🡪Adaptive Maintenance**: Modifying the software to accommodate changes in the environment, such as updates to operating systems or hardware.

**🡪Perfective Maintenance**: Enhancing the software by adding new features or improving performance based on user feedback.

**🡪Preventive Maintenance**: Making changes to prevent future issues and improve system reliability.

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

Ans:

Web applications:

🡪Run in a web browser and are platform-independent.

🡪Accessible from any device with an internet connection and a browser.

🡪Updates are applied on the server side, ensuring all users have the latest version.

🡪Typically use cloud storage or remote databases.

Desktop Application:

🡪Installed and run on a specific operating system (Windows, macOS, Linux).

🡪Limited to the device on which they are installed.

🡪Users must manually install updates, which can lead to version discrepancies.

🡪Often store data locally on the user's device.

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

Ans: Advantages of web applications over desktop applications include:

**🡪Accessibility**: Available from any device with internet access.

**🡪No Installation**: Users can access without installation.

**🡪Automatic Updates**: Always up-to-date without manual effort.

**🡪Cross-Platform Compatibility**: Works on various operating systems.

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

Ans: UI/UX design plays a crucial role in application development by ensuring applications are user-friendly, visually appealing, and effective in achieving user goals. It bridges the gap between technology and the user, creating a positive and engaging experience.

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

Ans: Native Apps:

🡪Built for single platform like Swift for iOS, Kotlin for Android.

🡪Performance is excellent, fast and responsive.

🡪User experience is seamless, platform specific UI.

🡪Higher cost and longer development time.

🡪Eg: Apple maps/ Google maps

Hybrid Apps:

🡪They have single codebase like React-Native, Flutter.

🡪Performance is good but can be slower for complex apps.

🡪User experience is uniform across platform.

🡪Lower cost and faster development time.

🡪Eg: Instagram

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

Ans: In system analysis, Data Flow Diagrams (DFDs) are crucial for visually representing how data moves through a system, highlighting its components, processes, and interactions. They offer a clear and concise way to understand a system's functionality and identify potential areas for improvement, making them essential for both technical and non-technical stakeholders.

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

Ans: **Desktop Applications**

**🡪Pros:** High performance, offline access, better security, deep system integration.

**🡪Cons:** Platform-dependent, manual updates, limited accessibility, higher cost.

**Web Applications**

**🡪Pros:** Platform-independent, easy access, automatic updates, lower cost.

**🡪Cons:** Requires internet, performance limitations, security risks, limited device integration.

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

Ans: Flowcharts help in programming and system design by visually representing processes, logic, and data flow, making it easier to plan, understand, and debug complex systems. They simplify communication, streamline problem-solving, and ensure efficient development.