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

A program is a set of instructions written in a programming language that a computer follows to perform a specific task or solve a problem. It functions by receiving input, processing that input through logical steps, and producing output. The computer executes the instructions in sequence, or sometimes in a loop, to carry out the program's goal.

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

The key steps involved in the programming process are:

**1. Problem Analysis**: Understand the problem or task that needs to be solved.

**2. Planning/Design**: Outline the logic and structure of the solution (e.g., flowcharts, pseudo code).

**3. Coding**: Write the program using a programming language.

**4. Testing**: Run the program to find and fix any errors or bugs.

**5. Debugging**: Identify and correct any issues or bugs in the code.

**6. Optimization**: Improve the code's performance or efficiency if needed.

**7. Deployment**: Make the program available for use.

**8. Maintenance**: Continuously update and fix the program based on feedback or issues.

**3.** **What are the main differences between high-level and low-level.**

The main differences between high-level and low-level programming are:

1. **Abstraction**: High-level languages are more abstract, closer to human language, making them easier to read and write. Low-level languages are closer to machine code, requiring more detailed and complex instructions.
2. **Portability**: High-level languages are more portable, meaning they can run on different systems with minimal changes. Low-level languages are often system-specific and less portable.
3. **Control**: Low-level languages give programmers more direct control over hardware and memory, while high-level languages focus on ease of use and productivity, sacrificing some control for convenience.
4. **Examples**: High-level languages include Python, Java, and C++, while low-level languages include Assembly and machine code.

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

In web communication, the **client** and **server** play distinct roles:

* **Client**: The client is typically the user's device or application (like a web browser) that sends requests to the server. It initiates the communication by requesting resources (such as web pages, images, or data) from the server. The client then receives and displays the requested content to the user.
* **Server**: The server is a powerful computer or system that stores and manages the resources (such as websites, databases, and files). It listens for incoming requests from clients, processes those requests, and sends back the appropriate response (such as HTML, data, or files) to the client.

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

The **TCP/IP model** is a conceptual framework used to understand and implement network communication between devices over the internet. It defines a set of protocols that allow different devices to communicate across a network. The model has four layers, each responsible for specific tasks in data transmission:

1. **Application Layer**:
   * **Function**: Provides network services directly to end-users or applications (like web browsers, email clients). It includes protocols such as HTTP (web browsing), FTP (file transfer), and SMTP (email).
2. **Transport Layer**:
   * **Function**: Responsible for reliable data transfer between two devices. The main protocols here are **TCP (Transmission Control Protocol)** and **UDP (User Datagram Protocol)**.
3. **Internet Layer**:
   * **Function**: Responsible for routing data packets across different networks. The primary protocol here is **IP (Internet Protocol)**, which handles addressing and routing.
4. **Network Interface Layer** (or Link Layer):
   * **Function**: Deals with the physical transmission of data over the network hardware (like Ethernet, WI-Fi, or fiber optics).

**6. Explain Client Server Communication.**

Client-server communication is a process where a **client** (usually a user's device or application) sends a request to a **server** (a remote machine or system) for data or services. The server processes the request and sends back the appropriate response. This communication typically happens over a network, like the internet, using protocols such as HTTP. The client makes requests (like loading a webpage), and the server handles these requests, returning the necessary information (like the webpage content).

**7. How does broadband differ from fiber-optic internet?**

**Broadband** is a general term for high-speed internet, while **fiber-optic internet** is a specific type of broadband that uses light signals through fiber cables, offering faster speeds and higher reliability.

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

**HTTP** is an unsecured protocol used for transmitting data over the web, while **HTTPS** is the secure version, using encryption (SSL/TLS) to protect data and ensure privacy during transmission.

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

Encryption secures applications by protecting sensitive data during transmission and storage, ensuring confidentiality, integrity, and preventing unauthorized access.

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

**System software** manages hardware and provides a platform for running applications (e.g., operating systems), while **application software** performs specific tasks for users (e.g., word processors, browsers).

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

Modularity in software architecture enhances maintainability, reusability, and scalability by breaking down a system into independent, manageable components.

**12. Why are layers important in software architecture?**

Layers in software architecture are important because they organize a system into manageable, modular components, promoting separation of concerns, maintainability, and scalability.

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

A development environment is crucial in software production as it provides the necessary tools, configurations, and resources for efficient coding, testing, debugging, and collaboration.

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

**Source code** is human-readable instructions written in high-level programming languages, while **machine code** is binary code that the CPU can directly execute.

**15. Why is version control important in software development?**

Version control is important in software development because it tracks changes, enables collaboration, and allows developers to manage and revert to previous code versions, ensuring code integrity and minimizing conflicts.

**16. What are the benefits of using Github for students?**

Github offers students benefits like version control, collaboration on projects, access to open-source code, cloud-based storage, and free tools for learning and building portfolios.

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

**Open-source software** is freely available for modification and distribution, while **proprietary software** is owned by a company or individual, with restrictions on usage, modification, and distribution.

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

GIT improves collaboration by allowing multiple developers to work on the same project simultaneously, track changes, merge updates, and manage different versions of the code efficiently

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

Application software in businesses helps streamline operations, improve productivity, and support specific tasks such as accounting, customer relationship management, and data analysis.

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

The main stages of the software development process are **requirements gathering**, **design**, **coding**, **testing**, **deployment**, and **maintenance**.

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

The requirement analysis phase is critical because it defines the project's scope, goals, and functionalities, ensuring the software meets user needs and reducing the risk of costly changes later.

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

Software analysis identifies system requirements, user needs, and potential challenges, guiding the design and development to ensure the final product meets its objectives and functions effectively.

**23. What are the key elements of system design?**

The key elements of system design include **architecture design**, **data modeling**, **interface design**, **component design**, and **security planning** to ensure the system is functional, efficient, and scalable.

**24. Why is software testing important?**

Software testing is important to identify and fix bugs, ensure functionality, improve quality, and verify that the software meets user requirements and performs reliably.

**25. What types of software maintenance are there?**

The types of software maintenance are **corrective** (fixing bugs), **adaptive** (updating for new environments), **perfective** (enhancing features), and **preventive** (avoiding future issues).

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

**Web applications** run in a browser and require an internet connection, while **desktop applications** are installed on a computer and typically work offline.

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

Web applications offer advantages like **cross-platform accessibility**, **easy updates**, **remote access**, and **no installation** required, while desktop applications may be limited to specific operating systems and require manual updates.

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

UI/UX design plays a crucial role in application development by ensuring the interface is user-friendly, visually appealing, and provides a seamless, efficient experience for users.

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

**Native mobile apps** are developed for a specific platform (iOS or Android) using platform-specific languages, offering better performance, while **hybrid apps** are built using web technologies and run across multiple platforms with a single codebase, offering easier development but potentially lower performance.

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

DFDs (Data Flow Diagrams) are significant in system analysis as they visually represent the flow of data within a system, helping to identify processes, inputs, outputs, and data storage, which aids in understanding and designing the system.

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

**Pros of desktop applications**: Faster performance, offline access, and better integration with system resources.  
**Cons**: Platform dependency, manual updates, and limited accessibility.

**Pros of web applications**: Cross-platform compatibility, easy updates, and remote access.  
**Cons**: Requires an internet connection, potential performance limitations, and less integration with system resources.

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

Flowcharts help in programming and system design by providing a clear, visual representation of the logic and flow of processes, making it easier to understand, analyze, and communicate system functionality and decision-making steps.