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

* A program is set of instructions for a computer to execute specific tasks. It contains code written in a programming language which may be interpreted, compiled or assembled into machine readable form and then executed.
* In programming, functions are reusable blocks of code that perform specific tasks, allowing for modularity and organization, making programs easier to understand, maintain, and debug. It is defined by a name, may have parameters and may return a value.

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

* Key steps involved in Programming process are Analyzing the Problem, Algorithm design, Flowchart, Coding, Debugging, Testing, Final Output, Documentation.

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

* The main difference between high-level and low-level programming languages lies in their abstraction level and readability. High-level languages are closer to human language, easier to learn and write, and more portable, while low-level languages are closer to machine code, offering greater control over hardware but requiring a deeper understanding of computer architecture.

1. Describe the roles of the clients and server in web communication.

* In web communication, clients request information or resources from servers, which then process those requests and deliver the requested data, following a request-response pattern.
* Clients: Initiate requests to servers for web pages, data, or services.

Display the content received from servers. Handle user interactions and input.

* Servers: Process requests from clients. Retrieve and send requested content. Execute server-side logic and applications. Store and manage data.

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

* The TCP/IP model is a fundamental framework for computer networking. It stands for Transmission Control Protocol/Internet Protocol, which are the core protocols of the Internet. This model defines how data is transmitted
* over networks, ensuring reliable communication between devices. It consists of four layers:
* Application Layer - This is the highest layer, where applications interact with the network. It is responsible for end-to-end communication and error-free delivery of data.
* Transport Layer (TCP/UDP) - It Provides reliable and efficient data delivery between applications on different hosts.
* Network/Internet Layer (IP) - It Handles the routing of data packets across networks, ensuring they reach the correct destination.
* Network Access Layer - It Deals with the physical transmission of data across the network, including hardware and physical protocols. This Layer represents a collection of applications that require network communication.

1. Explain Client Server Communication.

* In an Operating System, Client Server Communication refers to the exchange of data and Services among multiple machines or processes. A client requests a service or resource from a server, which then processes the request and sends a response back to the client, following a request-response pattern and using a common communication protocol.
* This Communication model is widely used for exchanging data among various computing environments like Distributed Systems, Internet Applications, and Networking Application communication. The communication between Server and Client takes place with different Protocols and mechanisms.

1. How does broadband differ from fibre-optic internet?

* Broadband is a type of internet service which provides high speed internet for that service. It can carry out wide-bandwidth transmissions that have the capability to carry multiple signals. While fiber -optic internet is a specific type of broadband that uses light signals transmitted through thin glass or plastic fibers , offering faster speeds and lower latency compared to traditional broadband technologies like DSL or cable.

1. What are the differences between HTTP &HTTPS protocals?

* The key difference between HTTP and HTTPS lies in security: HTTP is insecure, transmitting data in plain text, while HTTPS encrypts data using SSL/TLS, ensuring secure communication and protecting sensitive information.
* HTTP: - It uses port number 80 for communication. It is considered to be unsecure.
* HTTP works at Application Layer. In HTTP, Encryption is absent. HTTP does not require any certificates.
* HTTP does not improve search ranking. HTTP faster than HTTPS. It does not use data hashtags to secure data. In HTTP Data is transfer in plaintext.
* HTTP Should be avoided. HTTP Does not require SSL/TLS or Certificate. In HTTP Users are worried about their data.
* HTTPS: - HTTPS for Hyper Text Transfer Protocol Secure. HTTPS for Hyper Text Transfer Protocol Secure. It uses 443 port number for communication. It is considered as secure.
* HTTPS works at Transport Layer. Encryption is present in HTTPS. HTTPS needs SSL Certificates. It helps to improve search ranking. HTTPS slower than HTTP.
* HTTPS Requires SSL/TLS implementation with Certificates. In HTTPS Users are confident about the security of their data.

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

* Protecting Data in Transit.
* Securing Data at Rest.
* Encrypting Backups
* Securing Communications
* Protecting Financial Transactions
* Enhancing IoT Security

1. difference between system software and application software?

* System software manages computer hardware and provides a foundation for application software to run, while application software performs specific tasks for users. It is the type of software that is the interface between application software and the system. Low-level languages are used to write the system software.
* Application Software is the type of software that runs as per user request. High -level languages are used to write the application software.
* The main difference between System Software and Application Software is that without system softwar8e, the system can not run on the other hand without application software, the Low-level maintenance system always runs.

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

* Modularity in software architecture, which involves breaking down a system into smaller, independent modules, is crucial for improving maintainability, reusability, scalability, and testability.

1. Why are layers important in software architecture?

* Layers are important in Software architecture because it promotes modularity, separation of concerns, and maintainability, making software easier to develop, test, and modify over time.

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

* A development environment is crucial in software production because it provides a controlled space for developers to write, test, and debug code without impacting live users or systems, ensuring higher quality and stability before deployment.
* A software development environment can play a big role in the stability, reliability, and ultimate success of a software offering.

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

* Source code is human-readable instructions written in a programming language, while machine code is the binary instructions (0s and 1s) that a computer's CPU directly executes.
* Source code can be converted into machine code by a compiler or an assembler. The resulting executable is machine code ready for the computer. Alternatively, source code can be executed without conversion via an interpreter. An interpreter loads the source code into memory.

1. Why is version control important in software development?

* Version control is important in software development because it enables efficient tracking and management of code changes, facilitating collaboration, enabling easy rollback to previous versions, and ensuring a clear history of project modifications.

1. What are the benefits of using Github for students?

* GitHub is a popular online platform where developers can host and share their code. For students, using GitHub offers numerous benefits including access to free tools and resources through the GitHub Student Developer Pack, a platform for showcasing projects and building a portfolio, and opportunities for collaboration and learning from others.

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

* The core difference between open-source and proprietary software lies in ownership and access to the source code: open-source software allows users to freely view, modify, and distribute the code, while proprietary software is owned by a single entity and restricts access and modification.

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

* GIT improves software development team collaboration by enabling multiple developers to work on the same project concurrently, facilitating parallel development, and providing a robust system for tracking and managing changes, ultimately streamlining workflows and reducing conflicts.

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

* Application software plays a crucial role in businesses by automating tasks, improving efficiency, facilitating communication, and enabling better decision-making through tools like CRM, ERP, and project management software.

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

* The main stages of the software development process, often referred to as the Software Development Life Cycle (SDLC), typically involve planning, requirements analysis, design, coding/implementation, testing, deployment, and maintenance.

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

* The requirement analysis phase is critical in software development because it establishes a clear understanding of what the software should achieve, ensuring the final product meets stakeholder needs and expectations, and minimizing costly rework later in the development process.

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

* Software analysis, a crucial phase in the development process which is focuses on understanding and documenting the needs of stakeholders to ensure the software meets their requirements and business goals. It involves requirement analysis, which determines the tasks needed for functional software, and validates requirements to ensure smooth functioning.

1. What are the key elements of system design?

* Key elements of system design include architecture, database design, APIs and communication, caching, load balancing, security, scalability, and performance, along with redundancy and fault tolerance.

1. Why is software testing important?

* Software testing is crucial because it identifies bugs and errors early, ensuring software functions correctly, meets user needs, and delivers a reliable and secure experience, ultimately saving time, resources, and improving customer satisfaction.

1. What types of software maintenance are there?

* Software maintenance encompasses four primary types: Corrective (fixing bugs), Adaptive (adapting to changes), Perfective (improving functionality), and Preventive (preventing future issues).

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

* Web applications are accessed through a web browser and require an internet connection, while desktop applications are installed locally on a device and can function offline.

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

* Web applications offer advantages over desktop applications like accessibility from any device with an internet connection, ease of updates, and cross-platform compatibility, while also being potentially more cost-effective and requiring less maintenance.

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

* UI/UX design in app development is crucial as it enhances user engagement, improves user satisfaction, reduces frustration, provides a competitive advantage, increases app downloads and retention, and enables data-driven optimization.

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

* Native apps are built specifically for a particular operating system (iOS or Android) using its native language, while hybrid apps are built using web technologies (HTML, CSS, JavaScript) and run within a native app "container" allowing them to work across multiple platforms.

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

* Data Flow Diagrams (DFDs) are crucial in system analysis because they provide a visual, easily understandable representation of how data flows through a system, aiding in understanding, communication, and identifying potential issues or areas for improvement.

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

* Desktop applications offer faster performance and offline access, while web applications provide broader accessibility and easier updates, each with pros and cons depending on the application's needs.

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

* Flowcharts help in programming and system design by visually representing the sequence of steps and decisions, aiding in planning, understanding, and debugging complex processes, and facilitating communication among developers.