Module 1 – SE : Overview of IT Industry

Q-1 : what a program is and how it functions.
A-1 : A program is a set of instructions written in a programming language that tells a computer how to perform a specific task or function.

Function of Program are :

-Coding
-Compilation
-Execution
-Completion

Q-2 : What are the key steps involved in the programming process?
A-2 : The main stages of programming are analysis, design, coding, testing, and maintenance.

Q-3 : What are the main differences between high-level and low-level programming languages?
A-3 : High-level languages like Python and Java that are closer to human language, making them easier to code and debug.

Low-level languages like Assembly are closer to machine code, offering more control over hardware, but are harder to write and understand.

Q-4 : Describe the roles of the client and server in web communication.
A-4 : The client (like a browser) requests resources from a server, while the server processes these requests and sends back the requested data to display to the user.

Q-5 : Explain the function of the TCP/IP model and its layers.
A-5 : The Application layer prepares data,
The Transport layer segments it and ensures reliable delivery,
The Internet layer handles addressing and routing packets,
The Network Access layer manages the physical transmission.

Q-6 : Explain Client Server Communication.
A-6 : A model where a client device (like a web browser or mobile app) sends a request over a network to a server, which processes the request and sends a response back to the client.

Q-7 : How does broadband differ from fiber-optic internet?
A-7 : Broadband is a general term for high-speed internet, while fiber-optic is a specific, more advanced type of broadband that uses thin glass or plastic fibers to transmit data as light.

Q-8 : What are the differences between HTTP and HTTPS protocols?
A-8 : 1.HTTP is a protocol used which transfer hypertext over the Web.

-HTTPS is an extended version of the Hypertext Transfer Protocol (HTTP). It is used for secure communication.

2.HTTP sends data in plain text.
-HTTPS encrypts data using SSL/TLS protocols.

3.No encryption is applied to the data transmitted through HTTP.
-HTTPS employs encryption to secure data transmission.

4.HTTP does not verify the identity of the server.
-HTTPS verifies the server's identity using digital certificates.

5.HTTP is typically used for general website browsing.
-HTTPS is preferred for transmitting sensitive information, such as passwords or credit card details.

Q-9 : What is the role of encryption in securing applications?
A-9 : Encryption secures applications by protecting data from unauthorized access, loss, and theft through scrambling it into an unreadable format.

Q-10 : What is the difference between system software and application software?
A-10 : 1.System Software maintains the system resources and gives the path for application software to run.
-Application software is built for specific tasks.

2.Low-level languages are used to write the system software.
-While high-level languages are used to write the application software.

3.Example: System software is an operating system, etc.
-Example: Application software is Photoshop, VLC player, etc.

4.System Software programming is more complex than application software.
-Application software programming is simpler in comparison to system software.

Q-11 : What is the significance of modularity in software architecture?
A-11 : Modularity in software architecture is significant because it decomposes a system into independent, interchangeable components, which improves the software's flexibility, scalability, and maintainability.

Q-12 : Why are layers important in software architecture?
A-12 : Layers are important in software architecture because they promote modularity, separation of concerns, and maintainability by dividing an application into distinct parts with specific responsibilities.

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

A-13 : A development environment is crucial for software production because it provides a safe "sandbox" for developers to write, test, and experiment with code without affecting live users.

Q-14 : What is the difference between source code and machine code?
A-14 : Source code is human-readable code written by programmers, while machine code is a low-level, non-readable format of binary digits (0s and 1s) that a computer's processor can directly execute.

Q-15 : Why is version control important in software development?
A-15 : Version control systems (VCS) are important in software development for several key reasons:

-Collaboration and Teamwork,
-Tracking Changes and History,
-Rollbacks and Error Recovery,
-Branching and Experimentation,
-Code Quality and Review,
-Backup and Disaster Recovery,
-Project Management and Organization.

Q-16 : What are the benefits of using Github for students?
A-16 : Students benefit from GitHub through the Student Developer Pack, which provides free access to professional tools, learning resources, and cloud services.

Q-17 : What are the differences between open-source and proprietary software?
A-17 : The main difference is that open-source software has its source code publicly available, allowing anyone to use, study, modify, and distribute it,

while proprietary software is owned by a company, its source code is kept secret, and its use is restricted by a license.

Q-18 : How does GIT improve collaboration in a software development team?
A-18 : Git significantly improves collaboration in a software development team through several key features and a distributed workflow:

-Parallel Development with Branching
-Efficient Merging and Conflict Resolution
-Code Review and Feedback with Pull Requests
-Clear Version History and Accountability
-Distributed Nature.

Q-19 : What is the role of application software in businesses?
A-19 : Application software empowers businesses by automating tasks, improving productivity, and facilitating communication and collaboration.

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

A-20 : The main stages of the software development process are planning, requirements analysis, design, development (or coding), testing, deployment, and maintenance.

Q-21 : Why is the requirement analysis phase critical in software development?
A-21 : The requirement analysis phase is critical in software development because it ensures the final product meets user needs, which in turn minimizes costly rework, prevents misunderstandings, and mitigates risks early on.

Q-22 : What is the role of software analysis in the development process?
A-22 : The requirement analysis phase is critical in software development because it ensures the final product meets user needs, which in turn minimizes costly rework, prevents misunderstandings, and mitigates risks early on.

Q-23 : What are the key elements of system design?
A-23 : System design includes defining the architecture, components, modules, and interfaces of a system to meet specific requirements.

Q-24 : Why is software testing important?
A-24 : Software testing is important because it ensures the software is reliable, secure, and performs as expected, leading to higher user satisfaction and fewer costly issues.

Q-25 : What types of software maintenance are there?
A-25 : The four types of software maintenance are corrective (fixing bugs), adaptive (modifying for environmental changes), perfective (improving performance or adding features), and preventive (preventing future issues).

Q-26 : What are the key differences between web and desktop applications?
A-26 : The key differences are that web applications are accessed via a browser and require an internet connection but need no installation, are accessible from any device, and receive central updates.

Q-27 : What are the advantages of using web applications over desktop applications?
A-27 : Web applications offer advantages like accessibility from any device, centralized automatic updates, and platform independence, as they run in a browser and don't require installation.

Q-28 : What role does UI/UX design play in application development?
A-28 : UI/UX design plays a critical role in application development by ensuring an application is both visually appealing (UI) and easy to use (UX), which increases user satisfaction, engagement, and retention.

Q-29 : What are the differences between native and hybrid mobile apps?
A-29 : A native app is built specifically for a single mobile operating system using the platform’s official languages and tools (e.g., Swift/Objective C for iOS, Kotlin/Java for Android).

-A hybrid app is built using web technologies (HTML, CSS, JavaScript) or cross-platform frameworks, and then wrapped inside a native container so it can run on multiple platforms.

Q-30 : What is the significance of DFDs in system analysis?
A-30 : DFDs are significant in system analysis because they provide a clear, graphical representation of how data moves through a system, making complex processes easier to understand, improving communication between technical and non-technical stakeholders, and helping to identify inefficiencies.

Q-31 : What are the pros and cons of desktop applications compared to web applications?
A-31 : Desktop applications excel with better performance and offline access, but they require installation and lack cross-platform compatibility.

-Web applications are accessible from anywhere with an internet connection and are easy to update, but they often have slower performance and require an internet connection to function.

Q-32 : How do flowcharts help in programming and system design?
A-32 : Flowcharts help in programming and system design by providing a visual blueprint to map out processes, algorithms, and system structures, which aids in planning, debugging, and communication.