

Q1. What is software?

Ans: Software is a set of instructions, data or programs used to operate computers and execute specific tasks. It is the opposite of hardware, which describes the physical aspects of a computer. Software is a generic term used to refer to applications, scripts and programs that run on a device. It can be thought of as the variable part of a computer, while hardware is the invariable part.

Q1. What is software engineering?

Ans: Software Engineering is the process of designing, developing, testing, and maintaining software. It is a systematic and disciplined approach to software development that aims to create high-quality, reliable, and maintainable software. Software engineering includes a variety of techniques, tools, and methodologies, including requirements analysis, design, testing, and maintenance.

Q2. Explain types of software?

Ans: (1) Layers of Software- Let's explore software architecture layers with a lighthearted analogy: Imagine you're building a delicious and elaborate cake. Each layer of the cake represents a different aspect of software architecture. In this there are 5 topic to understand this mentioned below, 1. Presentation Layer 2. Application Layer 3. Business Layer 4. Persistence Layer 5. Database Layer

(2) Software Environment- In this software there are 5 variants mentioned below,

1. The Analysis and Design Environment

2. The Development Environment

3. The Common Build Environment

4. The Testing Environment

5. The Production Environment

(3) Application Software- Application software is a type of computer program that performs specific functions for the end-user by interacting directly with it. There are different types of application software, and here are some examples: Word Processing Software: This type of software is used to create, edit, and format text documents. Examples include Microsoft Word, Google Docs, and Apple Pages. Spreadsheet Software: This type of software is used to organize, analyze, and manipulate numerical data. Examples include Microsoft Excel, Google Sheets, and Apple Numbers. Presentation Software: This type of software is used to create visual presentations that can be displayed on a screen or projector. Examples include Microsoft PowerPoint, Google Slides, and Apple Keynote. Web Browsers: Web browsers are used to access and view websites on the internet. Examples include Google Chrome, Mozilla Firefox, and Microsoft Edge. Multimedia Software: This type of software is used to create, edit, and play multimedia content such as audio and video files. Examples include Adobe Photoshop, VLC

Media Player, and Windows Media Player. Educational Software: This type of software is designed to assist in teaching or self-learning.

(4) Software Architecture Software architecture refers to the high-level design and organization of a software system. It encompasses the overall structure, components, relationships, and principles that guide the development and maintenance of a software application. Software architecture provides a blueprint for building a reliable, scalable, and maintainable software solution. Here are some key aspects of software architecture:

1. **Structure and Components:** Software architecture defines the structure of a system and its components. It identifies the major building blocks, such as modules, classes, and components, and determines how they interact and collaborate with each other.

2. **Design Patterns:** Software architecture leverages design patterns, which are proven solutions to common design problems. Design patterns provide reusable templates and guidelines for solving recurring design challenges, ensuring that the software is designed in a robust and efficient manner.

3. **System Behavior and Interactions:** Architecture describes how different components of a system interact with each other and how the system behaves as a whole. It specifies the flow of data, control, and communication between various modules and subsystems.

4. **Quality Attributes:** Software architecture addresses various quality attributes or non-functional requirements, such as performance, reliability, scalability, security, and maintainability. It defines strategies and mechanisms to ensure that the software meets these requirements.

5. **Decisions and Trade-offs:** Architecture involves making important design decisions and trade-offs. It considers factors like performance versus cost, simplicity versus flexibility, and speed of development versus long-term maintenance. Architects evaluate different options and select the most suitable approaches based on project constraints and goals.

6. **Stakeholder Collaboration:** Software architecture requires collaboration with stakeholders including business analysts, developers, testers, and project managers. Architects communicate the architectural vision, solicit feedback, and ensure that the architecture aligns with business objectives and user needs.

7. **Evolution and Adaptability:** Software architecture accounts for the future evolution and adaptability of the system. It should be designed in a way that allows for easy integration of new features, technology upgrades, and scalability to handle increased user loads.

8. **Documentation and Communication:** Architecture documentation serves as a reference for developers, maintainers, and future teams working on the software. Clear and concise

documentation helps communicate design decisions, system behavior, and component interactions.

Overall, software architecture is a crucial aspect of software development that provides a structured approach to designing complex software systems. It facilitates collaboration, guides decision-making, and ensures that the resulting software solution meets the required quality attributes while being adaptable to future needs.

Q3. What is SDLC?

Ans: SDLC means Software development life cycle and there are 6 phases in that

1 Analysis- Analysis means product owner, Project manager, Business Analyst, CTO

2 Design- Design means System Architect, UX/UI designer

3 Development- Development means Front-end developer and Back-end developer

4 Testing- Testing means Solution Architect, QA Engineer, Tester, DevOps

5 Deployment- Deployment means Data Administrator, DevOps

6 Maintenance- Maintenance means Users, Testers, Support managers