MODULE: 1

1. What is software? What is software engineering?

ANS: Software refers to the set of programs, instructions, and data that operate a computer system and enable it to perform various tasks. It encompasses everything from the operating system that manages hardware resources to applications like word processors, games, and web browsers.
Software engineering, on the other hand, is a discipline within computer science that deals with the design, development, testing, and maintenance of software systems. It involves applying engineering principles and methodologies to software development to ensure that software products are reliable, efficient, maintainable, and scalable. Software engineering encompasses various processes, techniques, and tools aimed at managing the complexity of software systems and delivering high-quality software within time and budget constraints.

2. Explain types of software

ANS: Software can be classified into several types based on various criteria, including its purpose, functionality, and how it is distributed. Here are some common types of software:

1. System Software : This type of software is responsible for managing computer hardware and providing a platform for running application software. Examples include operating systems like Windows, macOS, Linux, and device drivers.

2. Application Software: Application software is designed to perform specific tasks or functions for end-users. It can range from productivity tools like word processors and spreadsheets to multimedia software, games, and web browsers.

3. Programming Software : Programming software includes tools and environments used by developers to create, debug, and maintain software applications. Examples include integrated development environments (IDEs), compilers, debuggers, and text editors.

4. Utility Software : Utility software serves various auxiliary functions to support the operation of computer systems and enhance user productivity. Examples include antivirus software, disk cleaners, backup software, and file compression tools.

5. Embedded Software : Embedded software is specialized software designed to control embedded systems or devices like microcontrollers, industrial machines, consumer electronics, and automotive systems. It is often tailored to specific hardware and performs dedicated functions.

6. Middleware : Middleware software acts as an intermediary between different software applications or components, facilitating communication and data exchange. Examples include web servers, database management systems, and messaging middleware.

7. Firmware : Firmware is a type of software that is closely tied to hardware and is typically stored in non-volatile memory. It provides low-level control over hardware components and is often used in devices like routers, printers, and electronic devices.

8. Open Source Software (OSS) : Open source software refers to software whose source code is freely available for anyone to inspect, modify, and distribute. Examples include the Linux operating system, the Apache web server, and the Mozilla Firefox browser.

3. What is SDLC? Explain each phase of SDLC

ANS: SDLC stands for Software Development Life Cycle. It is a systematic process used by software engineers to develop high-quality software efficiently. The SDLC consists of several phases, each with its own set of activities and deliverables. Here are the common phases of the SDLC:

1. Requirements Gathering and Analysis:

In this phase, the development team works closely with stakeholders to gather and document the requirements for the software. The team identifies stakeholders' needs, objectives, and constraints. Requirements are analyzed for feasibility, clarity, and completeness.

2. System Design:

Based on the requirements gathered, the system architecture and design are developed.

This phase involves creating a high-level design that outlines the overall structure and components of the system.

Detailed designs for individual components are also created, including database design, user interface design, and software architecture.

3. Implementation/Coding:

In this phase, the actual coding and programming of the software are done. Developers write code according to the design specifications. Best coding practices and standards are followed to ensure the code is readable, maintainable, and efficient.

4. Testing:

The testing phase involves verifying that the software meets the specified requirements and functions correctly. Different types of testing are conducted, including unit testing, integration testing, system testing, and acceptance testing. Defects and bugs are identified and fixed during this phase.

5. Deployment/Installation:
Once the software passes testing and is deemed ready for release, it is deployed to the production environment. Installation procedures are prepared and executed to install the software on users' systems or servers. Data migration, if necessary, is also performed during this phase.

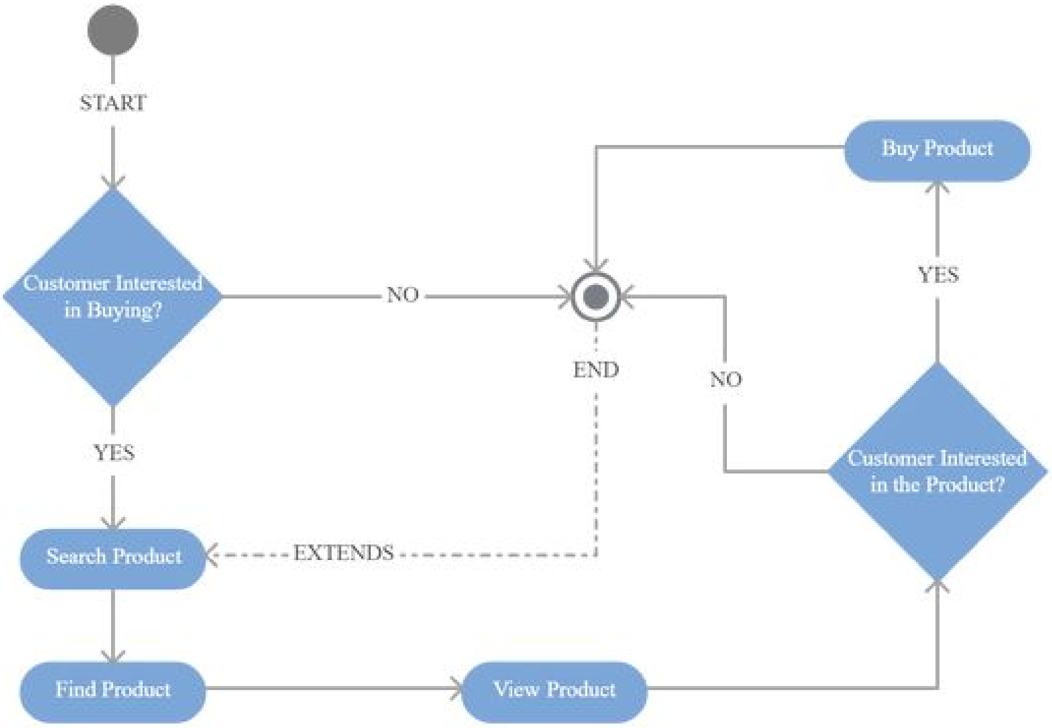
6. Maintenance and Support:
After deployment, the software enters the maintenance phase, where it is regularly monitored and updated as needed. Bug fixes, performance enhancements, and feature additions may be implemented based on user feedback and changing requirements.
Technical support is provided to address any issues that arise during the software's operational lifespan.

4. What is DFD? Create a DFD diagram on Flipkart

ANS: DFD stands for Data Flow Diagram. It is a graphical representation of the flow of data within a system, illustrating how data is input, processed, and output. DFDs are commonly used in software engineering and systems analysis to model the structure and behavior of systems.

Creating a DFD for Flipkart, an e-commerce platform, would involve identifying the main components of the system, such as users, the website/application, databases, and external entities like payment gateways. Here's a simplified DFD diagram for Flipkart:

MODULE: 1



In this diagram:

- The "Flipkart Website" represents the main interface through which users interact with the system.
- "User" represents the individuals accessing the website to browse products, add items to their cart, and make purchases.
- "Database" stores product information, user data, orders, and other relevant data.
- "Payment Gateways" facilitate secure transactions by processing payment information provided by users.

This DFD diagram illustrates the flow of data between the various components of the Flipkart system. Users interact with the website to browse products and make purchases, which involves retrieving data from the database and processing payments through external payment gateways.