**SE - ASSIGNMENT MODULE-1**

**OVERVIEW OF IT INDUSTRY**

1. **What is software? What is software engineering?**

**ANS :-**

1. Software :-

Software refers to a set of instructions, programs, or data that enable a computer or a system to perform specific tasks or functions. It includes all the intangible components of a computer system, such as programs, applications, and operating systems, as opposed to hardware, which comprises the physical components like the computer's central processing unit (CPU), memory, and storage devices.

2. Software Engineering :-

Software engineering is the systematic application of engineering principles, methods, and tools to the design, development, testing, and maintenance of software systems. It involves a structured and organized approach to software development, with the goal of producing high-quality, reliable, and efficient software that meets the specified requirements. Software engineering encompasses a range of activities, including requirements analysis, system design, coding, testing, debugging, maintenance, and documentation.

Key aspects of software engineering include:

* Requirements Engineering :- Gathering and defining the requirements for the software to be developed.
* Design :- Creating a blueprint or plan for the software based on the requirements.
* Implementation :- Writing the code based on the design.
* Testing :- Verifying that the software functions correctly and meets the specified requirements.
* Maintenance :- Making updates and modifications to the software to address issues, add features, or adapt to changing requirements.

1. **Explain types of software ?**

**Ans :-**

* System Software :-
  + perating Systems (OS) :- Examples include Windows, macOS, Linux, and Android. They manage hardware resources and provide essential services to other software applications.
  + Device Drivers :- These enable communication between the operating system and hardware devices such as printers, graphics cards, and peripherals.
* Application Software :-
  + Productivity Software :- Includes word processors (e.g., Microsoft Word), spreadsheets (e.g., Microsoft Excel), and presentation software (e.g., Microsoft PowerPoint).
  + Web Browsers :- Software like Chrome, Firefox, Safari, and Edge that allows users to access and interact with websites on the internet.
  + Graphics Software :- Tools for creating and editing images and graphics, like Adobe Photoshop or GIMP.
  + Media Players :- Software for playing multimedia files, such as VLC Media Player or Windows Media Player.
  + Database Management Systems (DBMS) :- Software like MySQL, Oracle, or Microsoft SQL Server for managing and organizing large sets of data.
* Development Software :-
  + Integrated Development Environments (IDEs) :- Tools like Visual Studio, Eclipse, or IntelliJ IDEA that provide a comprehensive environment for software development.
  + Compilers and Interpreters :- Programs that translate high-level code into machine-readable code (compilers) or execute code directly (interpreters).
* Security Software :-
  + Antivirus Software :- Programs like Norton, McAfee, or Avast that protect systems from malicious software and security threats.
  + Firewall Software :- Helps monitor and control network traffic to prevent unauthorized access.
* Utility Software :-
  + Disk Cleanup Tools :- Software that helps optimize disk space usage and improve system performance.
  + Backup Software :- Tools for creating and restoring backups of important data.
  + File Compression Software :- Examples include WinRAR or 7-Zip, which compress and decompress files to save storage space.
* Network Software :-
  + Network Operating Systems (NOS) :- Operating systems designed for network use, like Windows Server, Linux Server editions, or Novell NetWare.
  + Network Monitoring Tools :- Software used to monitor and analyze network performance.
* Embedded Software :-
  + Firmware :- Software that is embedded in hardware devices, such as the software running on a printer, router, or IoT devices.

**3 .What is SDLC? Explain each phase of SDLC ?**

**Ans :-**

SDLC (Software Development Life Cycle) :-

The Software Development Life Cycle (SDLC) is a structured process used by software developers to design, develop, test, and deploy high-quality software systems. It consists of a series of well-defined phases, each with specific goals and deliverables. The SDLC aims to produce software that meets customer expectations, is delivered on time and within budget, and is maintainable over its lifecycle.

The typical phases of SDLC are :-

* Requirements Gathering and Analysis :-
  + Goal :- Understand and document the needs and expectations of the end-users and stakeholders.
  + Activities :- Gather and analyze requirements through meetings, interviews, surveys, and document the functional and non-functional requirements.
* System Design :-
  + Goal :- Create a blueprint for the software system based on the gathered requirements.
  + Activities :- Design the system architecture, user interface, database structure, and overall system specifications. This phase also involves decisions on hardware and software infrastructure.
* Implementation (Coding) :-
  + Goal :- Translate the design into actual code.
  + Activities :- Write, compile, and test the code according to the design specifications. This phase includes coding, unit testing, and debugging.
* Testing :-
  + Goal :- Verify that the software meets the specified requirements and is free of defects.
  + Activities :- Conduct various testing activities, including unit testing, integration testing, system testing, and user acceptance testing (UAT). Identify and fix defects.
* Deployment :-
  + Goal :- Release the software for public use or within the organization.
  + Activities :- Deploy the software to production environments and make it available to end-users. This phase may also involve user training and documentation.
* Maintenance and Support :-
  + Goal :- Ensure the software remains functional and adapts to changing requirements over time.
  + Activities :- Address issues discovered after deployment, implement updates, and make enhancements as needed. This phase can include corrective, adaptive, and perfective maintenance.
* Iterative and Incremental Approaches :-
* Some SDLC models follow an iterative or incremental approach, where the development process is divided into smaller cycles. After each cycle, a portion of the software is delivered, providing tangible progress and the opportunity to incorporate feedback.

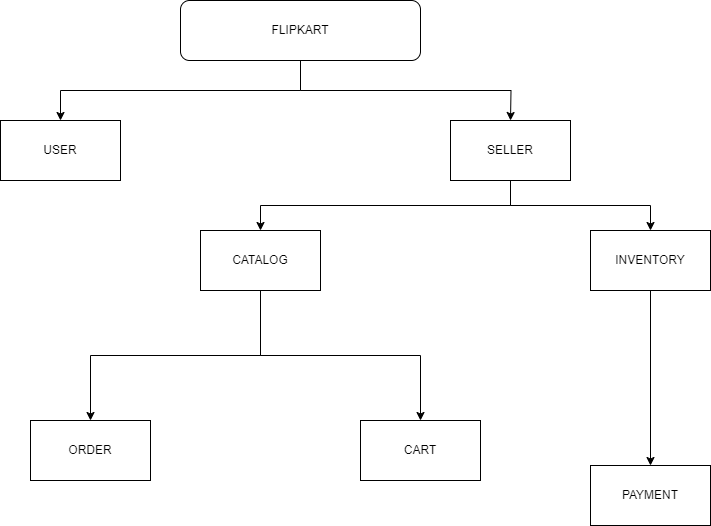
**4.What is DFD? Create a DFD diagram on Flipkart?**

**Ans :-**

DFD (Data Flow Diagram) :-

A Data Flow Diagram (DFD) is a visual representation that illustrates how data moves within a system and how processes operate on that data. It is a modeling technique used in software engineering and systems analysis to depict the flow of data through various processes and data stores in a system. DFDs provide a clear and concise way to understand the input, processing, and output of information within a system.

DFD diagram on Flipkart :-



**5. What is Flow chart? Create a flowchart to make addition of two numbers?**

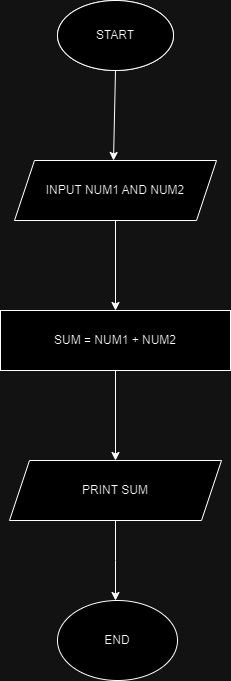
**Ans:-**

Flowchart :-

A flowchart is a visual representation of a process, showing the steps involved and the flow of control between them. It uses different shapes to represent different types of steps or processes and arrows to indicate the flow of control or data.

Flowchart Symbols:\*\*

* Start/End :-
  + Represents the beginning or end of a process.
* Process :-
  + Represents a task or operation.
* Decision :-
  + Represents a decision point where the flow can take one of two or more paths.
* Input/Output :-
  + Represents input from or output to the user or another system.
* Flow Arrows :-
  + Indicate the flow of control or data between different steps.
* Flowchart for Addition of Two Numbers :-



* **Explanation :-**
* Start :-
  + Represents the beginning of the process.
* Input Num1 :-
  + Represents the input of the first number.
* Input Num2 :-
  + Represents the input of the second number.
* Process: Add Num1 and Num2 :-
  + Represents the addition operation.
* Output: Result :-
  + Represents the output of the result after adding Num1 and Num2.
* End :-
  + Represents the end of the process.
* Flow :-
  + The flow follows a linear path, starting with input, performing the addition, and ending with the output.

**6. What is Use case Diagram? Create a use-case on bill payment on paytm.**

**Ans :-**

Use Case Diagram :-

A use case diagram is a visual representation of the interactions between different actors (users or systems) and the system itself. It illustrates the various ways in which users or external systems interact with a system to achieve specific goals or tasks.

Use Case Diagram for Bill Payment on Paytm :-

