**MODULE: 1**

**SE – Overview of IT Industry**

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

🡪 Software refers to the programs, data, and instructions that tell a computer what to do. It encompasses everything from operating systems like Windows or macOS to applications like word processors, web browsers, and video games.

🡪Software Engineering is a systematic approach to the design, development, maintenance, testing, and evaluation of software. The aim is to produce high-quality software that meets the needs of users efficiently and reliably.

**2.Explain types of software.**

a)System Software:

This includes operating systems (like Windows, macOS, Linux), device drivers, utilities, and more. System software serves as a base for application software and manages hardware.

b) Application Software:

These are programs that perform specific tasks for users, such as word processors (e.g., Microsoft Word), spreadsheets (e.g., Excel), media players, and more.

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

🡪 SDLC (Software Development Life Cycle) stands for Software Development Life Cycle. It is a structured process used by software developers to design, develop, test, and deploy software applications. The purpose of SDLC is to produce high-quality software that meets customer requirements within a specified timeline and budget. The SDLC process is usually divided into several phases:

a) Planning :

* In this stage, project goals, scope, budget, and timelines are defined.
* Identify resources, costs, timeframes, and potential risks.
* Project stakeholders collaborate to establish a clear understanding of the software's purpose and expected outcomes.

b) Requirements Analysis :

* Gather detailed business and system requirements.
* Conduct interviews, surveys, and meetings with stakeholders.
* Document functional and non-functional requirements.
* Create requirement specification documents (e.g., Software Requirement Specification or SRS).

c) Design :

* Transform requirements into a blueprint for the software.
* Develop system architecture and design.
* Designers create detailed blueprints and specifications that serve as a roadmap for the development team.

d) Implementation (or Coding) :

* Write the code to build the software based on the design documents.
* Use appropriate programming languages and tools.
* Follow coding guidelines and standards.
* Perform code reviews and unit testing.

e) Testing :

* Verify that the software works as intended and meets the requirements.
* Perform various types of testing (unit, integration, system, acceptance).
* Identify and fix defects and bugs.
* Ensure the software is reliable, secure, and performs well.

f) Deployment :

* After successful testing, the software is deployed to the production environment or released to end-users.
* Perform final testing and validation.
* Provide user training and documentation.

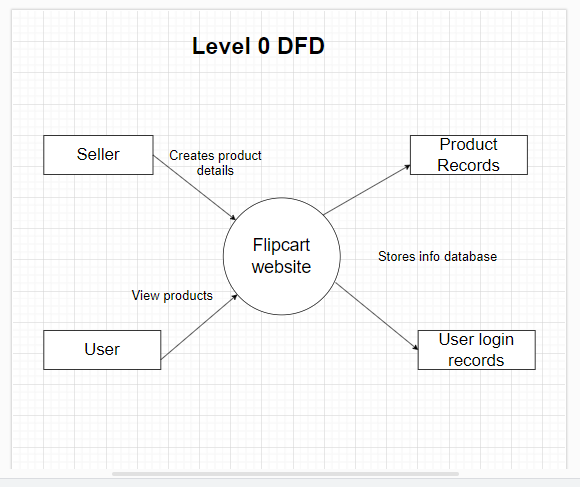
g) Maintenance :

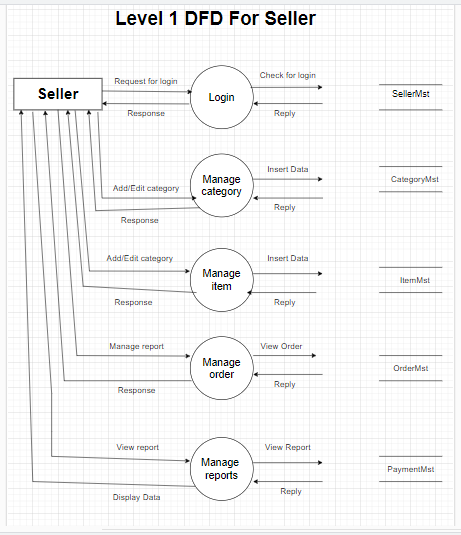
* Ensure the software continues to function correctly and meets user needs.
* Monitor software performance.
* Provide technical support and user assistance.
* Implement updates, bug fixes, and new features.
* Perform regular maintenance tasks.

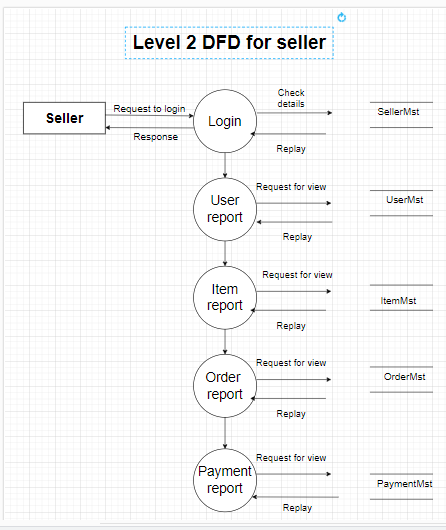
­­­­

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

🡪 A Data Flow Diagram (DFD) is a graphical representation used to visualize the flow of data within a system. It helps to understand how data moves from input to output through various processes and storage points.

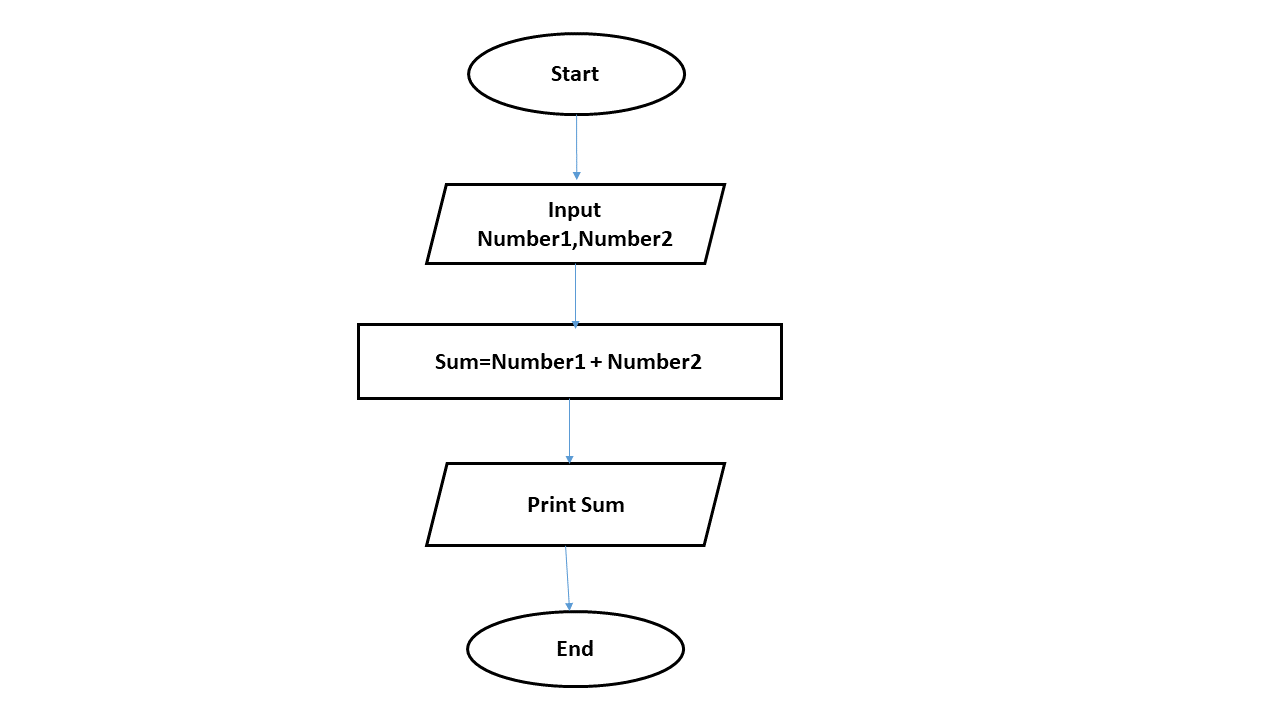






**5. What is Flow chart? Create a flowchart to make addition of two numbers.**

🡪A flowchart is a diagram that represents a process, system, or algorithm using various symbols and arrows to illustrate the sequence of steps and the flow of control or data. Flowcharts are commonly used in different fields, including computer programming, project management, business process modeling, and engineering, to help understand, analyze, and communicate processes clearly.



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

🡪 A Use Case Diagram is a graphical representation of the interactions between users (actors) and a system to accomplish specific goals or tasks. It's a tool used in software engineering to capture and describe the system's functionalities from a user's perspective.

