

### Q.1 What is software? What is software engineering?

**Ans** **software**, instructions that tell a computer what to do. Software comprises the entire set of programs, procedures, and routines associated with the operation of a computer system. The term was coined to differentiate these instructions from hardware—*i.e.*, the physical components of a computer system. A set of instructions that directs a computer's hardware to perform a task is called a program, or software program.

**Software Engineering** is an engineering branch related to the evolution of software product using well-defined scientific principles, techniques, and procedures. The result of software engineering is an effective and reliable software product.

### Q. Explain type of software

**Ans.** Types of software:

1>.general Purpose Software: A general purpose software application is software which has been designed to solve user's common or general tasks using a computer system. Examples include word processors, spreadsheets and photo editing applications for example

2>.customized Software: Customized software (also known as bespoke software or tailor-made software) is software that is specially developed for some specific organization or other user.

3>.Utility Software: Utility software is a program specifically designed to help manage and tune system or application software. It is used to support the computer infrastructure - in contrast to application software, which is aimed at directly performing tasks that benefit ordinary users.

### Q.3 what is SDLC? Explain each phase of SDLC

**Ans.** An SDLC (software development life cycle) is a big-picture breakdown of all the steps involved in software creation (planning, coding, testing, deploying, etc.). Companies define custom SDLCs to create a predictable, iterative framework that guides the team through all major stages of development.

SDLC stands for Software Development Life Cycle, which is a structured approach followed by software development teams to plan, design, develop, test, and deploy software applications. The SDLC consists of several distinct phases, each with its own objectives, deliverables, and activities. Here is an explanation of each phase of the SDLC:

**Requirements Gathering:** In this phase, the development team works closely with stakeholders, such as clients, users, and business analysts, to gather and document the requirements of the software application. This involves understanding the purpose, functionality, and scope of the software.

**Analysis and Design:** In this phase, the gathered requirements are analyzed, and the system is designed. The development team creates a detailed system design that includes architectural decisions, data structures, algorithms, and user interfaces. This phase helps in identifying potential challenges and determining the most effective solution.

**Implementation:** Also known as the coding phase, this is where the actual development of the software takes place. The design specifications are translated into programming code using chosen programming languages, frameworks, and development tools. This phase focuses on writing clean, maintainable, and efficient code.

**Testing:** In this phase, the developed software undergoes rigorous testing to identify defects, bugs, and other issues. Various testing techniques are employed, such as unit testing, integration testing, system testing, and user acceptance testing, to ensure that the software functions correctly and meets the specified requirements.

**Deployment:** Once the software has passed the testing phase and is deemed ready for release, it is deployed into the production environment. This involves installing the software on the target systems, configuring it, and ensuring its compatibility with the existing infrastructure.

**Maintenance:** After the software is deployed, it enters the maintenance phase. This phase involves monitoring the software, addressing user feedback, fixing any issues or bugs that arise, and making necessary updates and enhancements. Maintenance can be categorized into corrective maintenance (fixing defects), adaptive maintenance (adapting the software to changes in the environment), and perfective maintenance (improving software performance or usability).

These phases of the SDLC provide a systematic and structured approach to software development, ensuring that the software meets the desired requirements, is thoroughly tested, and can be effectively maintained throughout its lifecycle. It is important to note that these phases are often iterative, meaning that they may be revisited or repeated as needed during the development process.

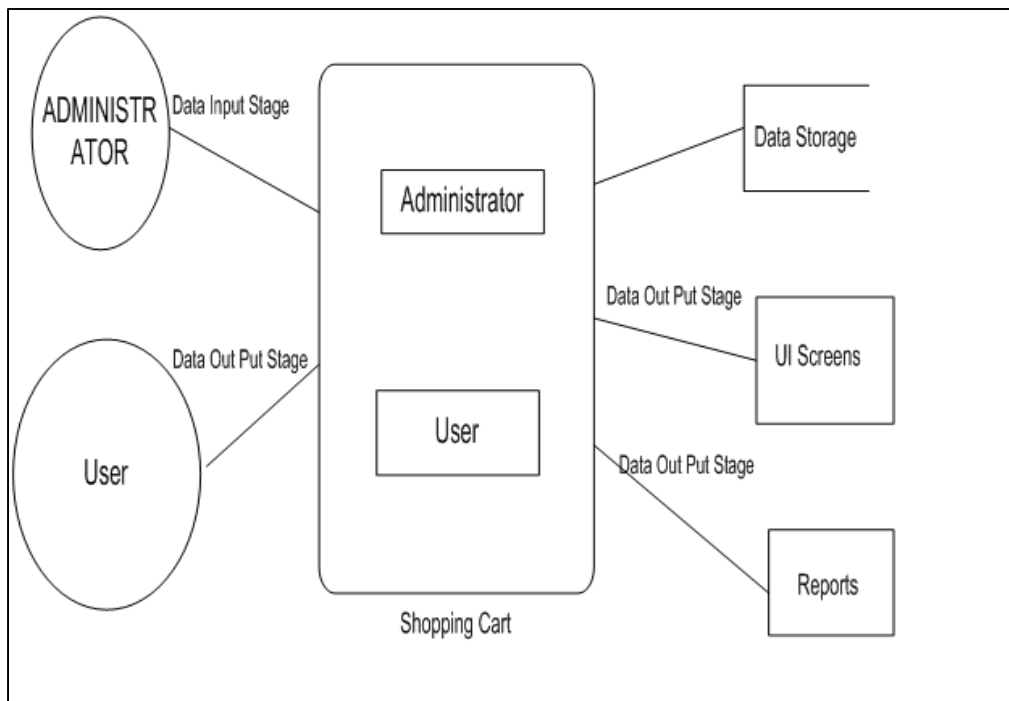
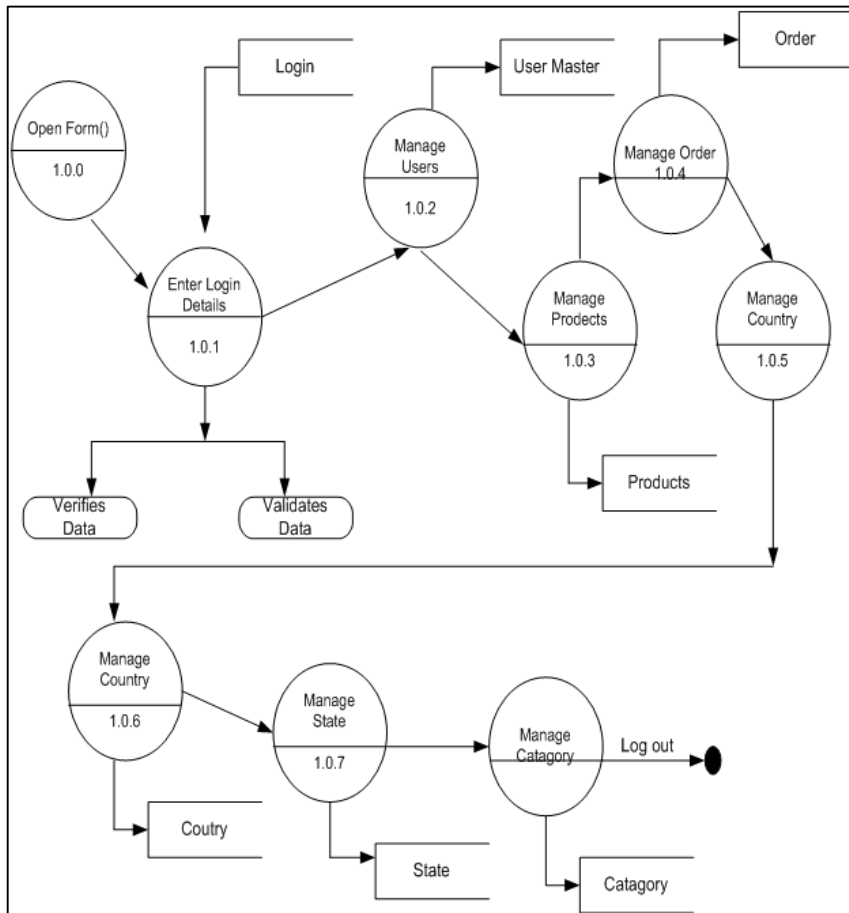
#### **Q.4 what is DFD? Create a DFD diagram on flip kart.**

**Ans.** DFD stands for Data Flow Diagram. It is a graphical representation that depicts the flow of data within a system or process. DFDs are widely used in software engineering and system analysis to visualize how data is input, processed, stored, and output by a system.

Creating a complete DFD diagram for a complex system like Flipkart, an e-commerce platform, would require a detailed understanding of its internal processes, data flows, and interactions.

However, I can provide you with a high-level overview of a DFD diagram for Flipkart, representing the major components and data flows.

Please note that this is a simplified representation and may not capture all the intricacies of the actual system. Here's an example:



**Customer:** Represents the users or customers of Flip kart who interact with the website and place orders.

**Website:** The interface through which customers interact with flipchart, browse products, add items to the cart, and place orders.

**Database:** Stores the product information, customer details, order history, and other relevant data.

**Inventory:** Manages the inventory of products, tracks stock availability, and updates the database accordingly.

**Payment:** Handles the payment processing, including interactions with payment gateways and financial institutions.

**Order:** Manages the order placement, order status, and updates the database with order-related information.

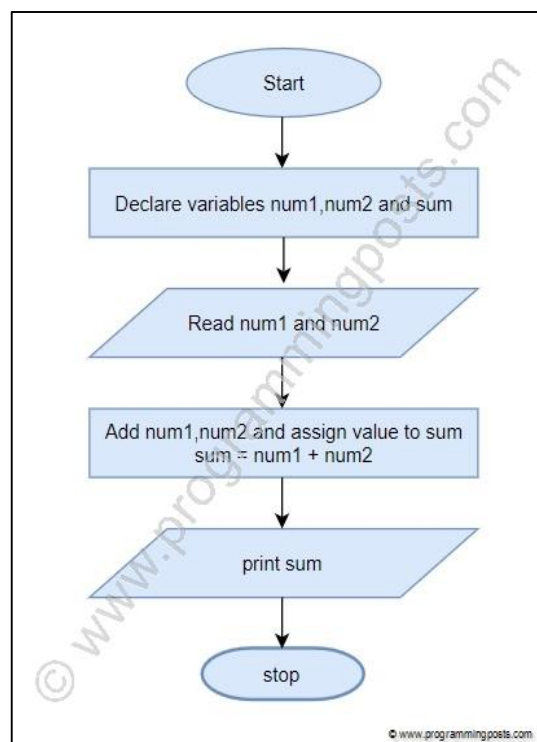
**Shipping:** Handles the shipping and logistics, including generating shipping labels, tracking orders, and updating order status.

This DFD diagram provides a simplified overview of the major components and data flows in flip kart. In a real-world scenario, the diagram would likely be more complex and include additional components and data flows.

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

Ans. A flowchart is a graphical representation of a process or algorithm that uses various symbols and arrows to depict the sequence of steps or actions. It is commonly used to illustrate the logic or flow of a program, making it easier to understand and analyze.

Here is a simple flowchart to illustrate the addition of two numbers:



In this flowchart, the symbols used are as follows:

Start/End: Represents the beginning and end points of the flowchart.

Input/Output: Represents the input and output actions, such as entering numbers or displaying results.

Process: Represents a calculation or operation performed on the data.

Arrows: Connect the symbols and indicate the flow or sequence of actions.

Please note that the above flowchart is a basic representation and does not include any error handling or validation.

## Q.6 what is Use case Diagram? Create a use-case on bill payment on paytm

**Ans.** A use case diagram is a visual representation that illustrates the interactions between actors (users or external systems) and a system in terms of the desired functionality or behavior. It helps to identify and understand the various use cases or actions that actors can perform within the system.

Here is an example of a use case diagram for the bill payment feature on Paytm:

