**ASSIGNMENT NO.1**

**SUBJECT: SOFTWARE ENGINEERING.**

**Q1. What is software? What is software engineering?**

**Ans:** A software is nothing but programs or set of instructions or set of information, set of dynamic or static data programs to operate computers or execute specific task or given task.

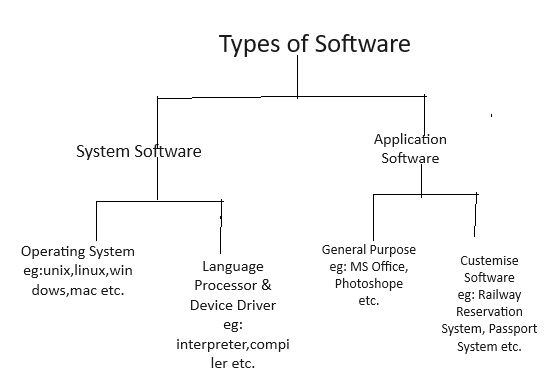
Software engineering is nothing but branch but also like to work discipline of designing, creating, and maintaining software systems using stablished engineering principles or engineering methods.

**Q2. Explain types of software?  
Ans:** The two main categories of software are application software and system software. An application is software that fulfills a specific need or performs a task. System software is designed to run a computer’s hardware and provides a platform for applications to run on top of.

**Application Software:**

Application software is nothing but software application its can be perform specific task or fulfills a specific given need. It’s can be collaborated by system software or given hardware. An application software is run on system software or interoperate with hardware and system software with each other.

**System Software:**

System software is nothing but software it is collaborated like work as bridge on hardware and Application software. It is intermediator to hardware and software vice-versa. A System software is run on hardware and provide specific task or work on hardware use. Its can be provide securely and managing a hardware properly provide specific work on hardware. A System software it manages hardware and software’s properly and securely. 

**Q3. What is SDLC? Explain each phase of SDLC?**

**Ans: SDLC- [ Software Development Life Cycle.]:**

SDLC stands for Software Development Life Cycle. It is a process used by software development teams to design, develop, and test high-quality software efficiently. The SDLC framework consists of a series of phases that provide a systematic way to create software.

**The typical phases in an SDLC include:**

**Planning:** In this phase, the project scope, requirements, feasibility, and overall plan for the software project are defined.

**Analysis:** During this phase, the requirements are gathered and analysed to understand what the software needs to do.

**Design:** The design phase involves creating a high-level design that outlines the architecture of the software and how different components will interact.

**Implementation:** In this phase, the actual coding of the software takes place. Developers write code based on the design specifications.

**Testing:** The software is tested in this phase to identify and fix any defects or bugs. Various testing methods are used to ensure the software functions as intended.

**Deployment:** Once the software has been tested and approved, it is deployed to the production environment for end-users.

**Maintenance:** After deployment, the software enters the maintenance phase where updates, fixes, and enhancements are made as needed to ensure its continued functionality.

The SDLC helps ensure that software projects are completed on time, within budget, and meet the needs of stakeholders. It provides a structured approach to software development that helps teams manage complexity and deliver high-quality software products.

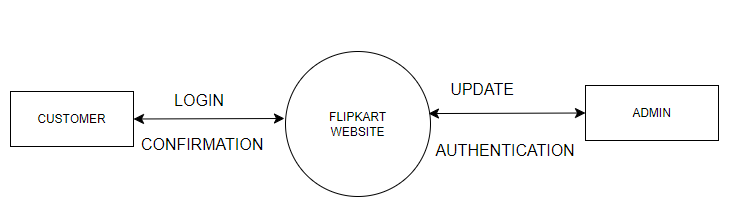
**Q4. What is DFD? Create a diagram on Flipkart?**

**Ans: DFD [Data Flow Diagram]:**

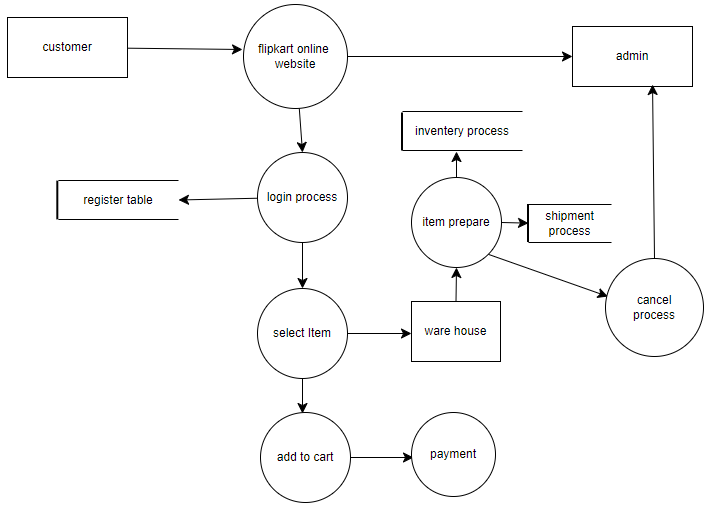
1. A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyse an existing system or model a new one.

Notation Steps:

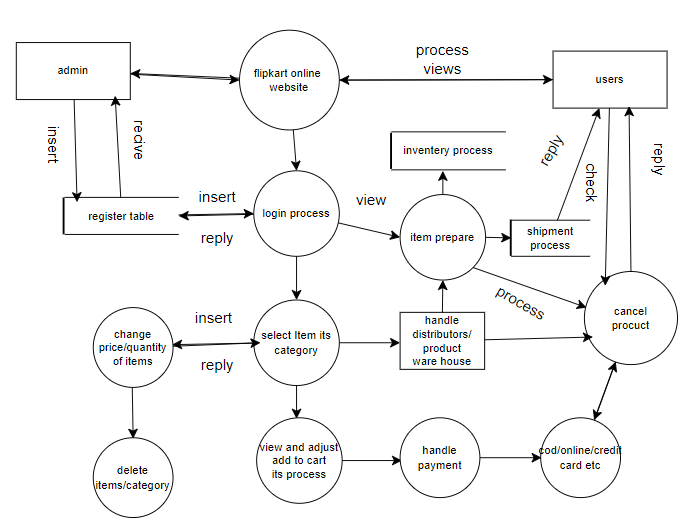
1. **External entity:** An outside system that sends or receives data, communicating with the system being diagrammed. They are the sources and destinations of information entering or leaving the system. They might be an outside organization or person, a computer system or a business system. They are also known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram.
2. **Process:** Any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules. A short label is used to describe the process, such as “Submit payment.”
3. **Data store:** Files or repositories that hold information for later use, such as a database table or a membership form. Each data store receives a simple label, such as “Orders.”
4. **Data flow:** the route that data takes between the external entities, processes and data stores. It portrays the interface between the other components and is shown with arrows, typically labelled with a short data name, like “Billing details.”

**** 0 LEVEL DFD-

1ST LEVEL DFD-



2ND LEVEL DFD-



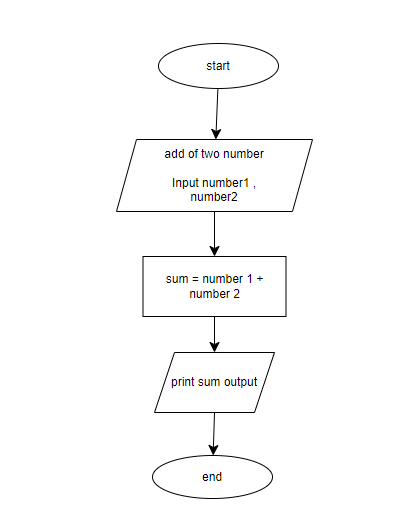
**Q5. What is Flow chart? Create a flowchart make addition of two numbers?**

**Ans:** A flowchart is a type of diagram that represents an algorithm, workflow, or process. The flowchart shows the steps as boxes of various kinds and their order by connecting the boxes with arrows.

It describes a process using symbols rather than words. Computer programmers use flow charts to show where data enters the program, what processes the data goes through, and how the data is converted to output.

**WHEN TO USE A FLOWCHART**

* To develop an understanding of how a process is done.
* To study a process for improvement.
* To communicate to others for a process is done.
* When better communication is needed between people involved with the same process.
* To document a process
* When planning a project



**Q6. What is Use case Diagram? Create use-case of on bill payment of Paytm?**

**Ans:** A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. ... The use cases, which are the specific roles played by the actors within and around the system.

Use case diagrams conceptually show the system's functional usage and their interrelationships, dependencies, extensions. etc. It also isolates the details of the system and enables us to understand it better and correlate the requirements with the fictional model as well.

In contrast, a well-structured list of use cases can be easier to understand and maintain. This approach allows you to focus on describing the individual functionalities in detail, without getting bogged down in the visual complexity of a diagram. The use case list can still capture the relationships between actors and use cases, but in a more textual and potentially more accessible format.

**Use-Case of On Bill Payment of Paytm**

