**MODULE: 1**

**SE – Overview of IT Industry**

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

**Ans:**

* **Software:**

Software is a collection of instructions, data, or programs that provides specific functionalities and controls the operation of a computer system. It is the intangible counterpart to hardware, which refers to the physical components of a computer.

* **Software Engineering:**

Software engineering is the systematic and disciplined application of engineering principles to the development, operation, and maintenance of software. It aims to create high-quality, reliable, and maintainable software that meets user needs effectively.

**Software engineers** are professionals who apply the principles of software engineering to design, build, test, and maintain software applications. Their work is crucial in various industries, driving innovation and enabling technological advancement.

**Q.2 Explain types of software**

**Ans:**

Software can be categorized into several different types, each serving a specific purpose:

1. **Application Software (Apps):**

* **Function:** Designed for **end-users** to perform specific tasks.
* **Examples:**
  + **Word processing:** Microsoft Word, Google Docs (create and edit text documents)
  + **Spreadsheet:** Microsoft Excel, Google Sheets (perform calculations, analyse data)
  + **Multimedia:** Media players, photo editors (play audio/video, edit images)
  + **Games:** Provide entertainment and interactive experiences
  + **Communication:** WhatsApp, Zoom (connect with others via text, video, audio)
  + **Web Browsers:** Chrome, Firefox (access websites and the internet)

**2. System Software:**

* **Function:** Works **behind the scenes** to manage hardware resources and provide a platform for other software to run.
* **Examples:**
  + **Operating Systems (OS):** Windows, macOS, Android, iOS (manage hardware, provide user interface)
  + **Device Drivers:** Act as translators between the OS and hardware devices (printers, graphics cards)
  + **Utility Software:** System maintenance, disk management, security tools (optimize performance, protect system)

**3. Middleware:**

* **Function:** Acts as a **bridge** between application and system software, facilitating communication and data exchange.
* **Examples:**
  + **Database management systems:** Manage and access information stored in databases
  + **Application servers:** Host and manage application software for multiple users
  + **Message queues:** Transfer data messages between different software systems

**4. Driver Software:**

* **Function:** A specific type of **system software** that allows the operating system to communicate with and control connected devices.
* **Examples:**
  + Printer drivers for managing printing tasks
  + Scanner drivers for capturing digital images
  + Network card drivers for enabling internet connection

**5. Programming Software:**

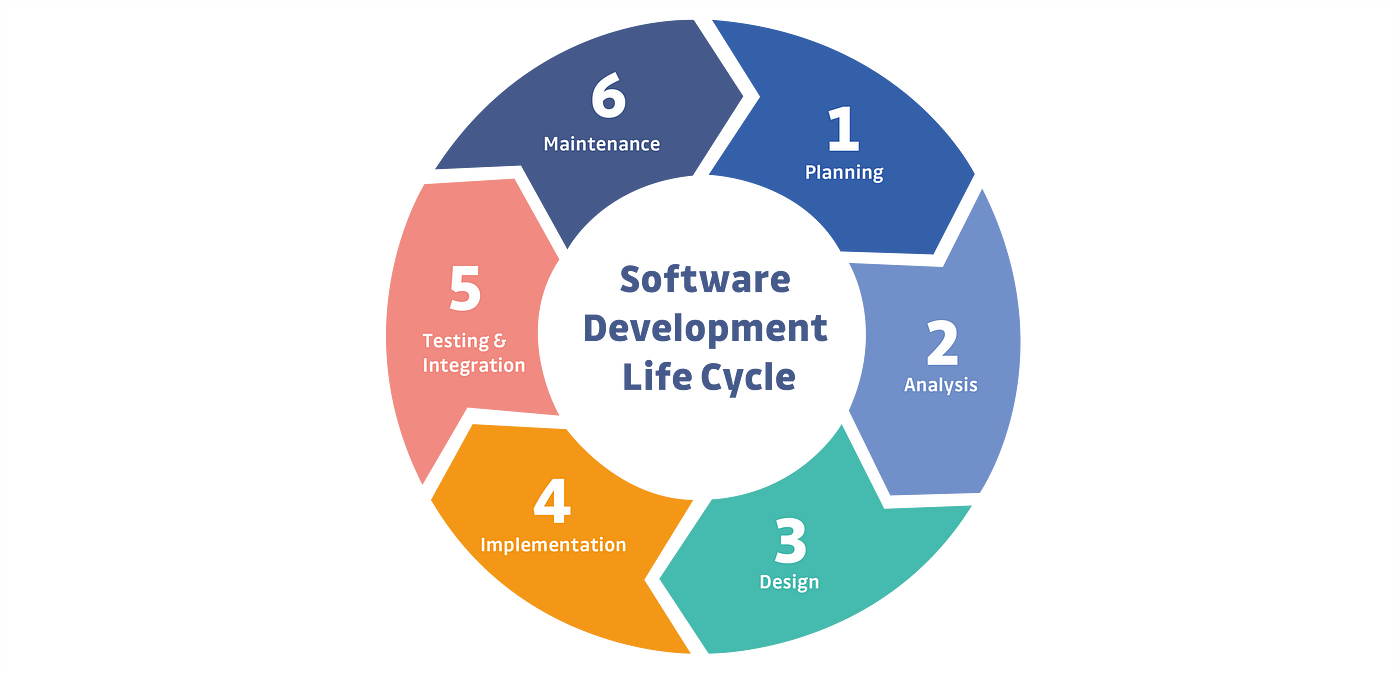
* **Function:** Used by **programmers** to develop other software applications.
* **Examples:**
  + **Integrated development environments (IDEs):** Provide tools for writing, testing, and debugging code (Visual Studio, Eclipse)
  + **Compilers:** Translate programming language code into machine code understandable by the computer

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

**Ans:**

* **What is SDLC?**

**SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.**



* **The Software Development Life Cycle (SDLC): 7 Phases**

There are several different models of the SDLC, but they all typically include the following stages:

1. **Planning:** This stage involves identifying the goals and objectives of the software, as well as the requirements of the users. The planning stage also involves creating a project plan, which outlines the tasks that need to be completed and the resources that will be needed to complete them. This stage is crucial as it sets the foundation for the rest of the project, if the goals or requirements are not well defined, the whole project will suffer.
2. **Analysis:** In this stage, the requirements of the users are analysed in more detail. This includes creating a detailed specification of the software, as well as determining the feasibility of the project. This stage helps to understand if the software is technically and economically viable and it helps to identify the resources and tools needed, and the time frame for development.
3. **Design:**The design stage involves creating a detailed plan for how the software will be developed. This includes designing the user interface, as well as the architecture and components of the software. It is essential to have a clear design that outlines the software’s structure and how the different parts will interact, it also allows the team to anticipate and solve technical issues before the implementation stage.
4. **Implementation:** During the implementation stage, the software is actually developed. This includes writing the code, as well as testing and debugging. This is the stage where the software starts taking shape, the code is written and put together, and the software starts to work as intended.
5. **Testing:** In this stage, the software is tested to ensure that it meets the requirements of the users and that it is free of bugs. There are several types of testing that can be done at this stage, such as unit testing, integration testing, and user acceptance testing. It is essential to have a comprehensive testing phase, as it will help identify issues and bugs early in the process, and avoid costly mistakes.
6. **Deployment:** After the software has been tested and found to be stable, it is deployed to the users. This stage can involve installing the software on a user’s computer or deploying it to a server. It is crucial to have a clear deployment strategy in place, to make sure the software reaches the end users in a smooth and controlled manner.
7. **Maintenance:**Once the software is deployed, it will require maintenance in order to fix any bugs that are discovered, as well as to make improvements and enhancements. This stage is ongoing and its purpose is to keep the software updated, secure, and functional. Software maintenance is important for ensuring the longevity of the software and for maintaining user satisfaction.

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

**Ans:**

# **DFD (Data Flow Diagram):**

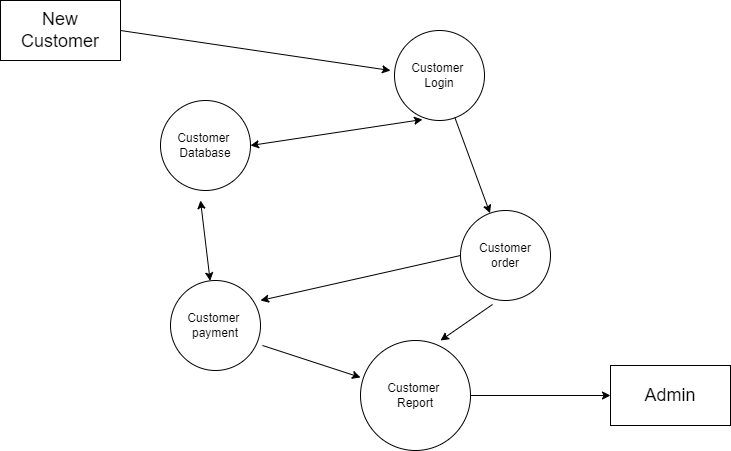
Data flow diagram (DFD) is a diagram being used frequently in software design. It visually represents the flow of data throughout processes in a given system. DFD shows the kind of information that will be input to and output from processes as well as where the data will be stored.

# **DFD Diagram on Flip Kart:**

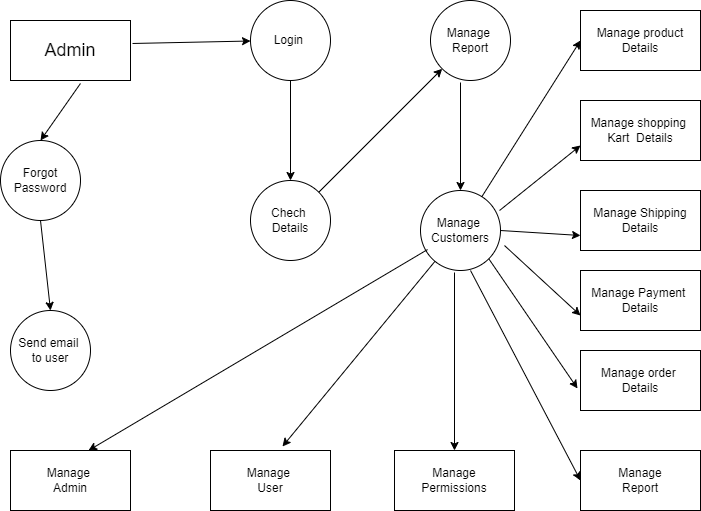
# **Level 0:**



# **Level 1:**



# **Level 2:**



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

**Ans:**

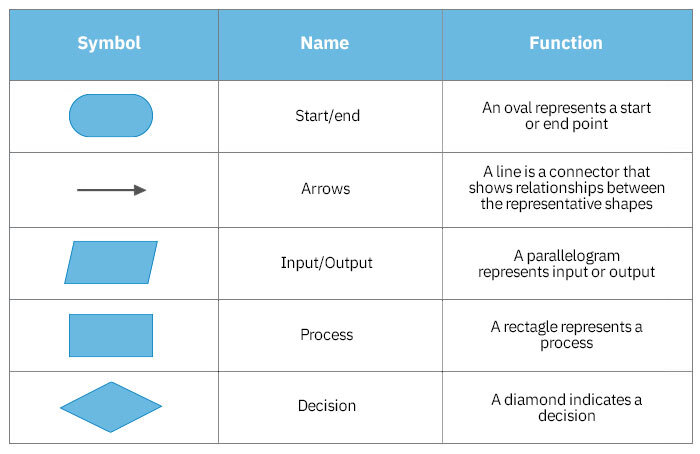
## Algorithm:

## The word [Algorithm](https://www.geeksforgeeks.org/fundamentals-of-algorithms/) means” *A set of finite rules or instructions to be followed in calculations or other problem-solving operations”*

* **Flowchart :**

Flowchart is a graphical representation of an algorithm. Programmers often use it as a program-planning tool to solve a problem. It makes use of symbols which are connected among them to indicate the flow of information and processing.

* **Basic Symbols used in Flowchart Designs :**



* **Flowchart to make addition of two numbers :**

****

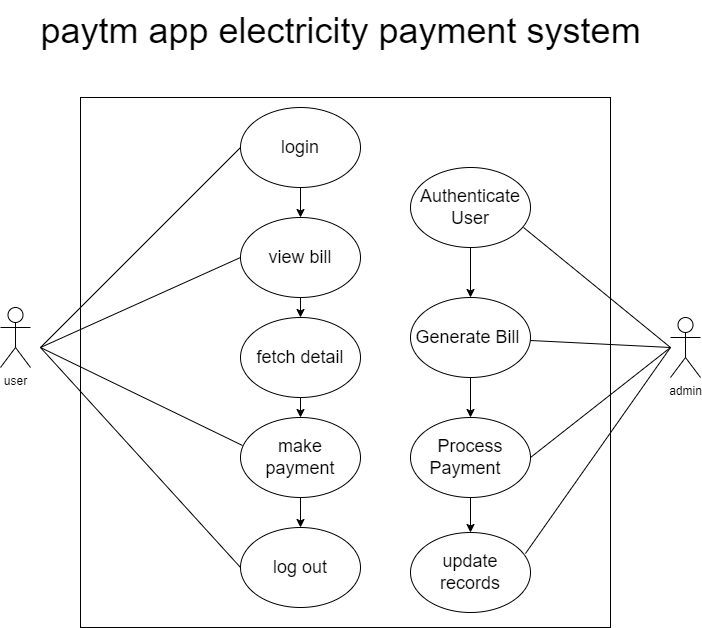
**Q.6**. **what is Use case Diagram? Create a use-case on bill payment on paytm.**

**Ans:**

* **What is a Use Case Diagram :**

A Use Case Diagram is a type of Unified Modelling Language (UML) diagram that represents the interaction between actors (users or external systems) and a system under consideration to accomplish specific goals. It provides a high-level view of the system’s functionality by illustrating the various ways users can interact with it.

* **Use Case Diagram on billing system :**

****