1. What is software? What is Software Engineering?

➢ Software:-

- Software is a set of instructions, data or programs used to operate computers and execute specific tasks.
- It is consists of lines of code written in programming languages.
- Software can be categorized into different types, including system software (e.g., operating systems, device drivers) and application software (e.g., word processors, web browsers, games). Essentially, software is what enables computers and other digital devices to carry out various tasks and provide functionality to users.

➢ Software Engineering:-

- Software Engineering is the process of developing software, maintaining, managing, testing and analysis.
- Software engineers use various techniques, methodologies, and tools to manage the complexity of software development and produce high-quality software products.
- It also involves considerations for factors such as software scalability, security, usability, and maintainability.
- Software engineers collaborate with stakeholders, such as clients, users, and project managers, to understand requirements and deliver software solutions that meet their needs.

2. Explain Type of Software.

➢ Meaning of Software:-

- "***Software is a set of programs (sequence of instructions) that allows the users to perform a well-defined function or some specified task.***"

➢ Type of Software

1. Application Software
2. System Software
3. Driver Software
4. Middleware
5. Programming Software

- **Application software:-**
  - The most common type of software, application software is a computer software package that performs a specific function for a user, or in some cases, for another application. An application can be self-contained, or it can be a group of programs that run the application for the user. Examples of modern applications include office suites, graphics software, databases and database management programs, web browsers, word processors, software development tools, image editors and communication platforms.
- **System software:-**
  - These software programs are designed to run a computer's application programs and hardware. System software coordinates the activities and functions of the hardware and software. In addition, it controls the operations of the computer hardware and provides an environment or platform for all the other types of software to work in. The OS is the best example of system software; it manages all the other computer programs. Other examples of system software include the firmware, computer language translators and system utilities.
- **Driver software:-**
  - Also known as device drivers, this software is often considered a type of system software. Device drivers control the devices and peripherals connected to a computer, enabling them to perform their specific tasks. Every device that is connected to a computer needs at least one device driver to function. Examples include software that comes with any nonstandard hardware, including special game controllers, as well as the software that enables standard hardware, such as USB storage devices, keyboards, headphones and printers.
- **Middleware:-**
  - The term *middleware* describes software that mediates between application and system software or between two different kinds of application software. For example, middleware enables Microsoft Windows to talk to Excel and Word. It is also used to send a remote

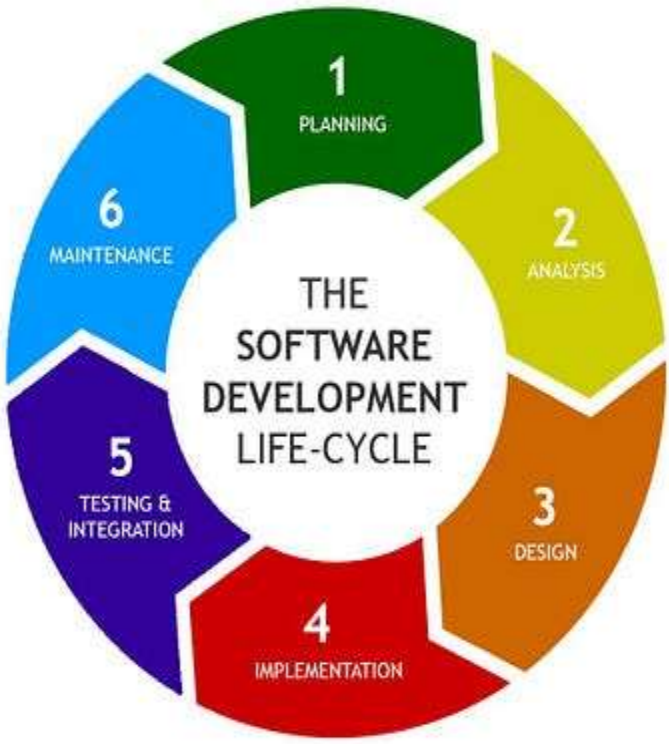
work request from an application in a computer that has one kind of OS, to an application in a computer with a different OS. It also enables newer applications to work with legacy ones.

- **Programming software.**
  - Computer programmers use programming software to write code. Programming software and programming tools enable developers to develop, write, test and debug other software programs. Examples of programming software include assemblers, compilers, debuggers and interpreters.

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

- **Meaning of SDLC:-**

The Software Development Life Cycle (SDLC) is a process used by software development organizations to plan, design, develop, test, deploy, and maintain software applications.



1) Planning

Business prerequisites are acknowledged in this phase.
Meetings with supervisors, partners and customers are held to decide the prerequisites like:

- Who is going to use the software application?
- How is the software application going to be used?
- What information is the software going to process?

These are general inquiries that get replied during the prerequisites acknowledgment stage.
A Requirement Specification document is then created to serve the purpose of guideline for the next phase of the cycle. It usually consists of the following:

- Functional Requirement Specification
- Business Requirement Specification
- Client/Customer Requirement Specification
- User Requirement Specification
- Business Design Document
- Business Document

2) Analysis

- In this phase, the project team works closely with stakeholders, including clients, users, and business analysts, to understand the software requirements. The goal is to define the scope of the project, identify features and functionalities needed, and document clear and detailed requirements.

3) Design

- In this phase, the high-level design of the software is created. This includes defining the architecture, data structures, user interfaces, and other components. The design aims to provide a blueprint for developers to follow during the coding phase.

4) Implementation (Coding)

- This is the phase where actual coding takes place based on the design specifications. Developers write the source code, following programming best practices and adhering to coding standards. This phase transforms the design into a functional software application.

5) Testing and Integration

- Although this stage is called testing, in real life, the situation is that faults found in the software in the testing phase, lead back to the development phase and then back to the testing phase in circles, until the software finally reaches the necessary quality. The testing phase normally consists of two internal phases:
- Automated unit/functional tests
- Acceptance tests performed by a human

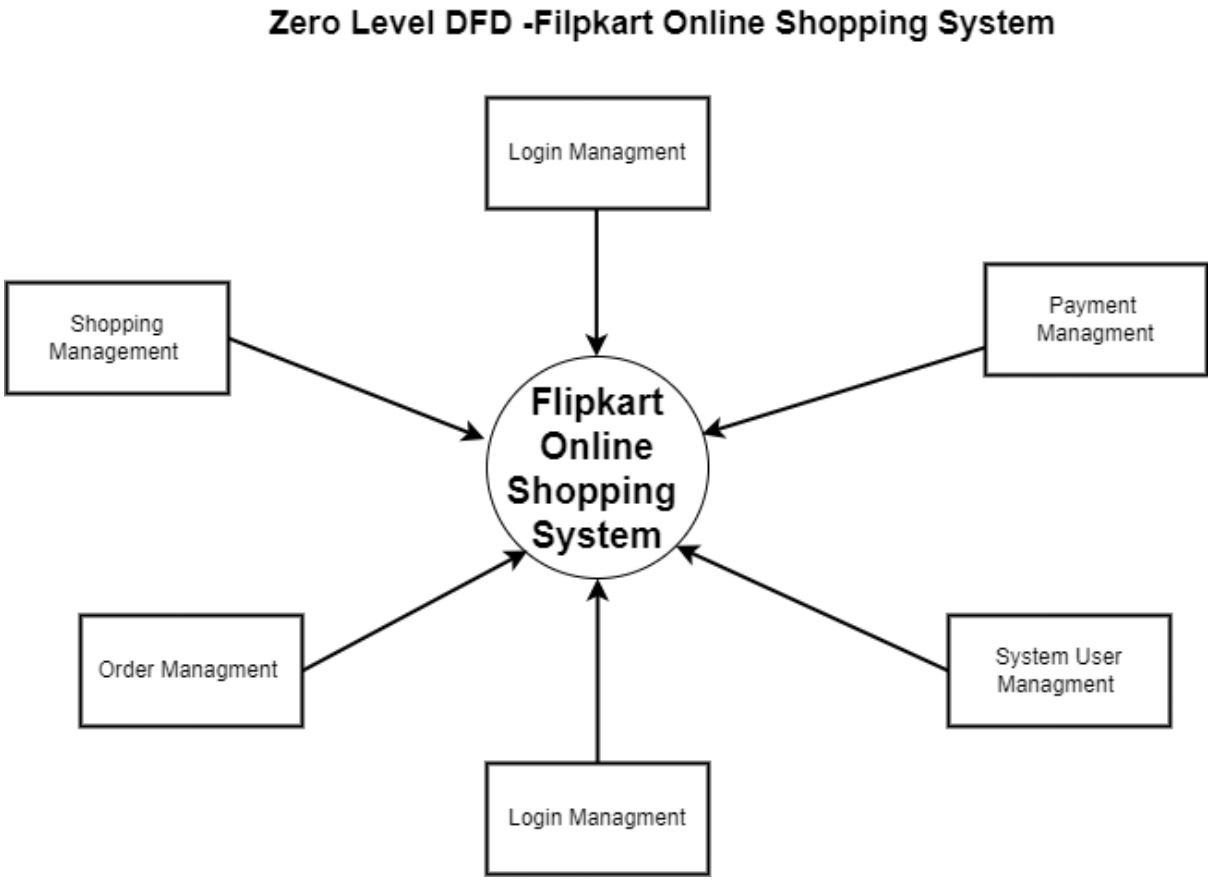
6) Maintenance

- Ignoring the technical side of this phase, this is when the software is released to the alpha/beta or stable state and feedback starts to come in. Analysis of the feedback can then lead to second third or fourth iterations of the SDLC which means, the process is continued again with phases 1, 2, 3, 4 and 5, gradually eliminating any issues that have been found in the released software.
- The whole software development process can be also be planned over several iterations, especially, when it comes to Agile, where the main focus is to release a working product as soon as possible and implement different features later on.
- So, it doesn't mean that going over several iterations always means bug-fixing or tuning. As I just mentioned — it can also be a pre-planned process.

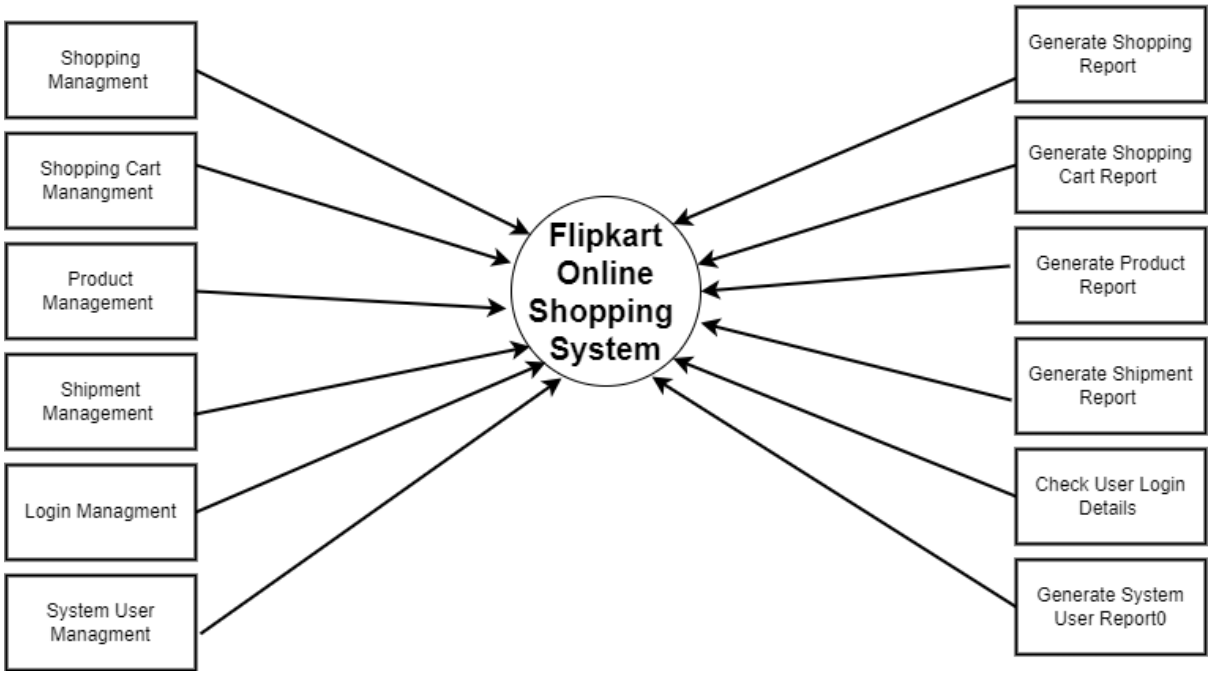
4. What is DFD? Create a DFD diagram on Flipkart.

- **Meaning of Flowchart:-**

DFD is the abbreviation for **Data Flow Diagram.** The flow of data of a system or a process is represented by DFD.



First Level DFD -Filpkart Online Shopping System

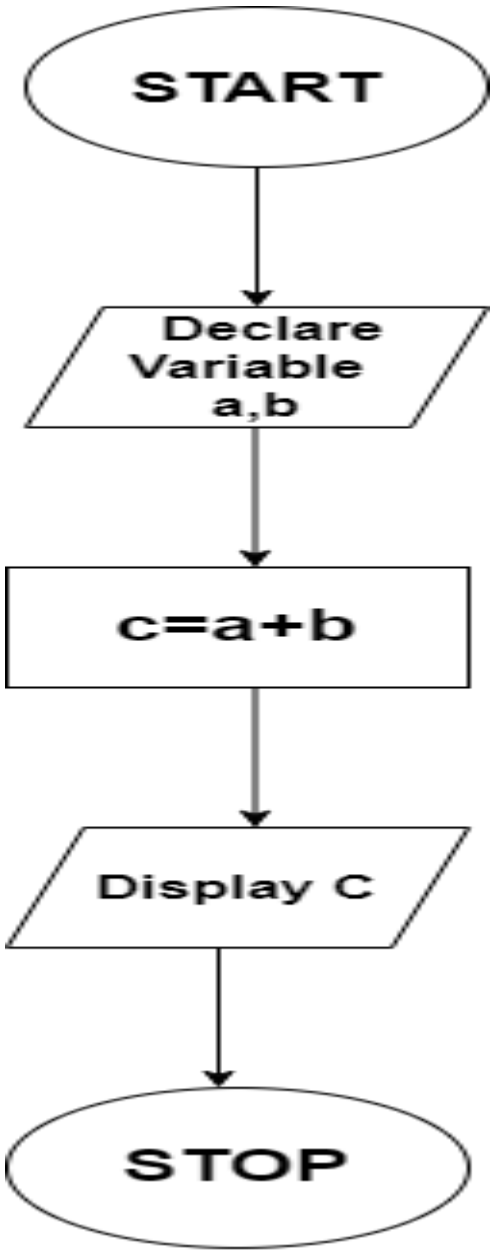


5. What is Flow Chart? Create a Flowchart to make addition of two number.

- **Meaning of Flowchart:-**

  A flowchart is a diagrammatic representation of sequence of logical steps of a program.Flowcharts use simple geometric shapes to depict process and arrows to show relationships and process data.

Addition of two number:-



6. What is use case diagram? Create a use-case on bill payment on paytm

- **Meaning of Case Diagram:-**

  A use case is a description of the ways in which a user interacts with a system or product. A use case may establish the success scenarios, the failure scenarios, and any critical variations or exceptions. A use case can be written or made visual with the help of a use case model tool.

- **Paytm bill Paytm:-**