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 the opposite of hardware, which describes the physical aspects of a computer. Software is a generic term used to refer to applications, scripts and programs that run on a device.

Software Engineering: Software engineering is defined as a process of analyzing user requirements and then designing, building, and testing software application which will satisfy those requirements.

1. Explain types of software?

Application Software

This is the most common type of computer software, and can be defined as end-user programs that help you perform tasks or achieve a desired outcome. The end-user is the person who is actually using a product or program. (They are the one for whom the “end result” is designed.) Some examples of application software include internet browsers, a CRM tool like Hubspot, a photo-editing software like Adobe or Lightroom, or a word processing application like Microsoft Word. Application software is installed on a computer or mobile device based upon a user’s need. Because this is the most common type of software, there are many options available and users can choose the one that best fits their needs, budget, and expectations. (For example, anyone wanting to look on the internet could use Chrome, Safari, or even Firefox.)

System Software

System software helps the user, the computer or mobile device, and an application all work together seamlessly. This makes system software crucial to running any kind of application software as well as the whole computer system.

Think about when your laptop or phone has an update. This is system software in action: there is a tweak made to the system software that helps your computer or phone continue to work well and keep applications running. Apple’s iOS is an example of system software, as is Microsoft Windows. System software is always running in the background of your device, but it is never something you will use directly. In fact, the only time most people remember it’s there is when it is time for an update.

Programming Software

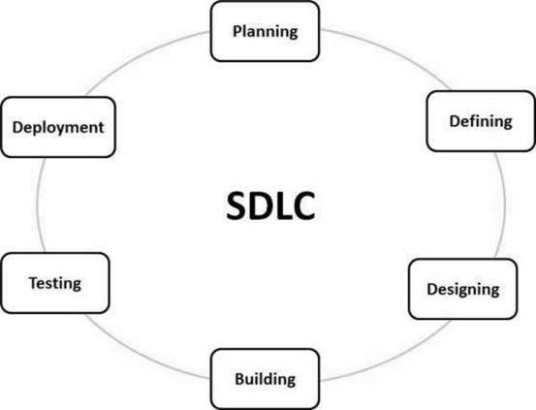
While application software is designed for end-users, and system software is designed for computers or mobile devices, programming software is for computer programmers and developers who are writing code. These are programs that are used to write, develop, test, and debug other software programs. It’s helpful to think of these programs as a translator of sorts: they take programming languages like Laravel, Python, C++, and more and translate them into something a computer or phone will understand.

### Driver Software

### This software is often considered to be a type of system software. Driver software operates and controls devices that are plugged into a computer. These drivers make it possible for devices to perform their necessary functions. A very good (and practical) example of this is your printer. When you are first setting up your printer to work with your computer, you have to install software to connect the two so that they communicate and print anything you need.

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

The software development lifecycle (SDLC) is the cost-effective and time-efficient process that development teams use to design and build high-quality software. The goal of SDLC is to minimize project risks through forward planning so that software meets customer expectations during production and beyond. This methodology outlines a series of steps that divide the software development process into tasks you can assign, complete, and measure.



### Plan

### The planning phase typically includes tasks like cost-benefit analysis, scheduling, resource estimation, and allocation. The development team collects requirements from several stakeholders such as customers, internal and external experts, and managers to create a software requirement specification document.

The document sets expectations and defines common goals that aid in project planning. The team estimates costs, creates a schedule, and has a detailed plan to achieve their goals.

### Design

In the design phase, software engineers analyze requirements and identify the best solutions to create the software. For example, they may consider integrating pre-existing modules, make technology choices, and identify development tools. They will look at how to best integrate the new software into any existing IT infrastructure the organization may have.

### Implement

In the implementation phase, the development team codes the product. They analyze the requirements to identify smaller coding tasks they can do daily to achieve the final result.

### Test

The development team combines automation and manual testing to check the software for bugs. Quality analysis includes testing the software for errors and checking if it meets customer requirements. Because many teams immediately test the code they write, the testing phase often runs parallel to the development phase.

### Deploy

When teams develop software, they code and test on a different copy of the software than the one that the users have access to. The software that customers use is called production, while other copies are said to be in the build environment, or testing environment.

Having separate build and production environments ensures that customers can continue to use the software even while it is being changed or upgraded. The deployment phase includes several tasks to move the latest build copy to the production environment, such as packaging, environment configuration, and installation.

### Maintain

In the maintenance phase, among other tasks, the team fixes bugs, resolves customer issues, and manages software changes. In addition, the team monitors overall system performance, security, and user experience to identify new ways to improve the existing software.

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

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.

### Purpose of Data Flow Diagrams

A typical information system involves processing a lot of information and processes. The purpose of Data Flow Diagrams is to view systems as a whole with its scopes and boundaries while it illustrates the movement of information between components. The focus of DFD is on the flow of data throughout the system, not process flow. DFD allows readers to easily see how the system will operate by knowing the kind and flow of information involved.

### DFD Types

There are two main types of DFD: Physical and Logical.

Physical DFD: Focuses on the physical aspect of the system by showing “how” the system will be implemented

Logical DFD: Focuses on a higher-level view of the system and “what” it will achieve.

### Data Flow Diagram Levels

Unlike other diagrams, DFD can be drawn at different levels, based on the purpose they are drawn to serve.

#### **Context Data Flow Diagram**

Context DFD is sometimes referred to as level 0 DFD. It’s the top-level diagram among all, which illustrates the entire system in its relationship to any external entities.

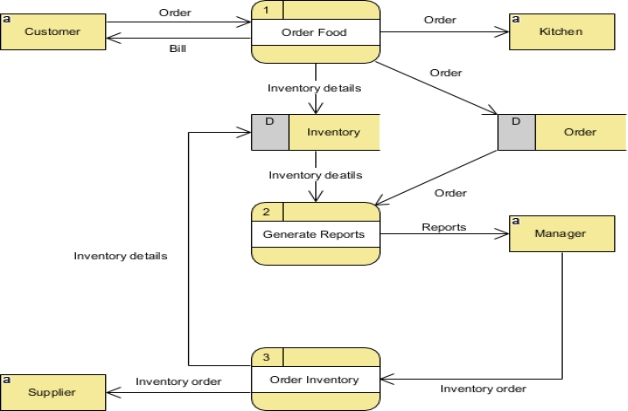
#### **Data Flow Diagram Level 1**

Level 1 DFD is the level under the context-DFD. It illustrates the main functions within the system. Level 1 breakdown the context level by including more details. It represents how the data enters and exits the system, where it is stored and how the basic processes convert it from one form to another.

#### **Data Flow Diagram Level 2**

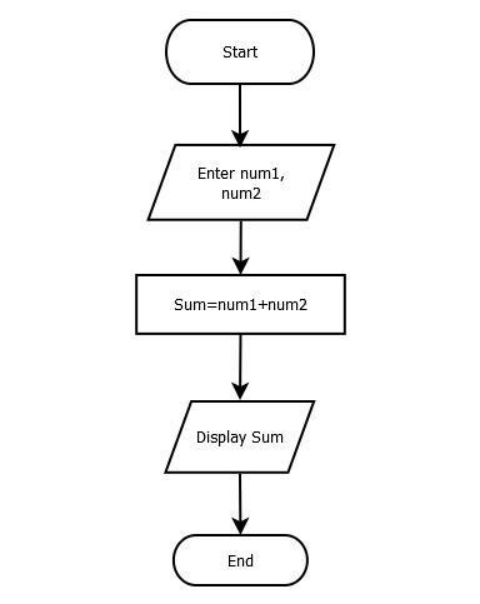
Level 2 DFD (or further) goes into deeper detail. It shows how the data flows within the main process of the system.

You should create as many levels of DFD as you need until the desired level of detail is met.



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

A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams. Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence.



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

Use Case Diagram for Payment. A use case diagram example developed for a Payment system. Use this design as a use case diagram example for teaching. The design can also be customized as a use case diagram template, with Visual Paradigm's use case diagram