

## **Q.1 – What is Software? What is Software engineering?**

### **What is 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. It can be thought of as the variable part of a computer, while hardware is the invariable part.

The two main categories of software are application software and system software. An application is software that fulfills a specific need or performs tasks. System software is designed to run a computer's hardware and provides a platform for applications to run on top of.

### **What is 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.

## Q.2 - Explain types of software

- Application software
- System software
- Driver software
- Middleware
- 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.

**Example:** Microsoft Office, Paint, Powerpoint etc..

# 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.
- 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..

**Example:**Notepad ,Calculator etc..

# 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.

**Example:** Audio Driver,Video Driver etc..

## 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.

**Example:** database middleware,application server middleware

## 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.

**Examples:** Turbo c,Eclipse,Sublime etc..

### Q.3 - What is SDLC? Explain each phase of SDLC

#### What is SDLC:-

- The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software.

- The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software. in detail, the SDLC methodology focuses on the following phases of software development:

1. Requirement Gathering

2. Analysis

3. Designing

4. Implementation

5. Testing

6. Maintenance



### **1. Gathering requirements**

At the first stage, we try to identify the answers to such questions: “What does the client need? What can be done for this?” “Should we create a new product or renovate the existing one?” We brainstorm and get maximum information from all interested parties. It’s important for the development team to understand the deadlines and what exactly the software should accomplish.

## **2. Defining**

This is a buffer stage between planning and acting when we analyze the requirements and ideas. Similar to planning a vacation, when you need to lay out your belongings and think about what to take with you, the defining phase helps save your time and headaches in the future. It is like a green light for your project.

## **3. Design**

During the design stage, our developers create a “skeleton” of the project based on the requirements. They define the technologies used, tools, workflows, relationships between different parts of the project, database structures, data flows, etc. As a result, we create a software requirements specification document, a roadmap of your project, - a thorough description of what and how to do from a technical point of view. This is an ongoing document where we add everything we do. It helps our team to see if we are on the right track before we start realizing the project. At this stage, we agree on all details and provide a final price and timeline for the project.

## **4. Development**

This is the lengthiest part of SDLC, but not the most difficult provided that the previous steps were done correctly. Once the development team gets the final requirements' specification, our software developers proceed to create the actual software. The coding tasks are delegated within the team depending on their area of expertise.

## **5. Testing**

Does the software we've built actually work? That's the question our team asks throughout the whole software development process. At Civity, we test every single component of the system to make sure

that all parts work together properly. If any bug comes up, the task is forwarded back to the development team. Testing is an ongoing process that continues until the software is free of bugs.

## **6. Deployment**

This is the phase when the world meets your product. The free of bugs software is sent to market for beta testing. It allows checking whether the final product meets the requirements and has room for improvements from the end-user perspective. Deployment can also occur in several stages when we release only the parts of your product to check if they are workable. Our team collects feedback, fixes bugs if found. After that, the final version of the product is rolled out.

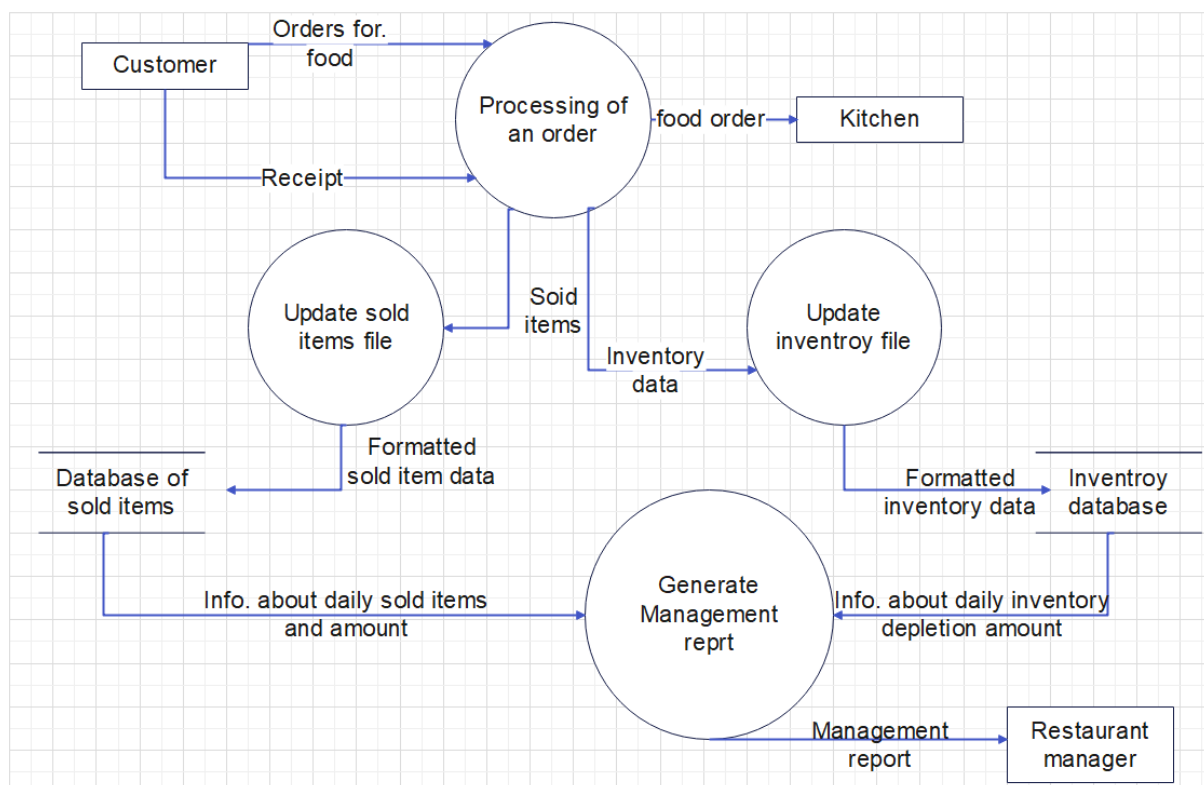
## **7. Maintenance**

It may seem that the developer's work ends at this stage, but this is not true. The created product needs to be updated periodically to ensure efficient use. If you have ambitious goals and want to improve your web product constantly, our team is ready to help you maintain the system and implement all needed updates. Maintenance is not the final stage of the software development life cycle. If the product continues to develop, the cycle starts over again.



#### Q.4 - What is DFD? Create a DFD diagram on Flipkart

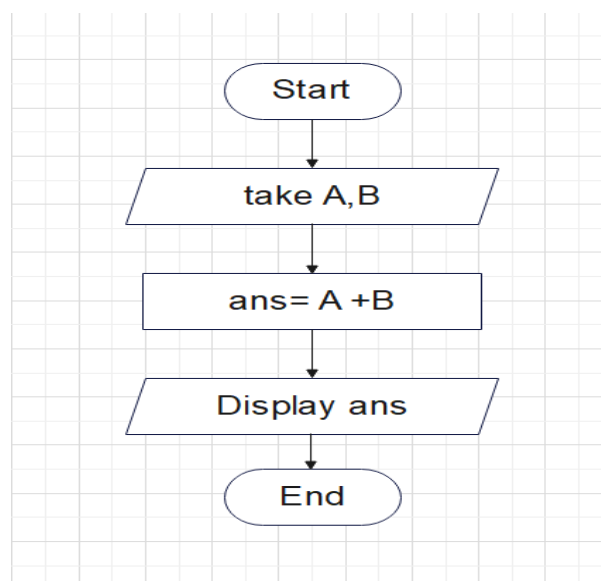
DFD is the abbreviation for Data Flow Diagram. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. Data Flow Diagram can be represented in several ways. The DFD belongs to structured-analysis modeling tools. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes.



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

Flowcharts are nothing but the graphical representation of the data or the algorithm for a better understanding of the code visually. It displays step-by-step solutions to a problem, algorithm, or process. It is a pictorial way of representing steps that are preferred by most beginner-level programmers to understand algorithms of computer science, thus it contributes to troubleshooting the issues in the algorithm. A flowchart is a picture of boxes that indicates the process flow in a sequential manner. Since a flowchart is a pictorial representation of a process or algorithm, it's easy to interpret and understand the process. To draw a flowchart, certain rules need to be followed which are followed by all professionals to draw a flowchart and is widely accepted all over the countries.

Create a flowchart to make addition of two numbers:-



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

A use case diagram is a way to summarize details of a system and the users within that system. It is generally shown as a graphic depiction of interactions among different elements in a system. Use case diagrams will specify the events in a system and how those events flow, however, use case diagram does not describe how those events are implemented.

