***Software Engineering Assignment***

**MODULE: 1**

***SE – Overview of IT Industry***

1.What is software? What is software engineering?

Ans. **Software** - Software is a set of instructions, data or programs used to operate computers and execute specific tasks.

**Software engineering** - Software engineering is defined as a process of analyzing user requirements and then designing, building, and testing software application

2. Explain types of software?

Ans. ● **Application software**: This type of software is designed to perform specific tasks for end-users. Examples include word processors like Microsoft Word, web browsers like Google Chrome, and graphic design tools like Adobe Photoshop. Application software serves a diverse range of purposes and is created to address the needs and preferences of users.

**● System software**: - System software is software that directly operates the computer hardware and provides the basic functionality to the users as well as to the other software to operate smoothly. Or in other words, system software basically controls a computer’s internal functioning and also controls hardware devices such as monitors, printers, and storage devices, etc. It is like an interface between hardware and user applications.

● **Driver software**: -Driver software, often referred to as device drivers, is a type of system software that facilitates communication between the operating system and hardware components or peripherals. These drivers act as translators, enabling the operating system to understand and utilize the functionalities of specific hardware devices. Whether it's a printer, graphics card, or network adapter, each hardware component requires a corresponding driver to ensure proper interaction with the operating system.

● **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?

Ans. SDLC, or Software Development Life Cycle, is a structured framework that outlines the stages involved in the development and maintenance of software. The SDLC aims to produce high-quality software that meets or exceeds customer expectations, is delivered on time and within budget, and is easy to maintain and enhance.

The following phase of SDLC.

1.**Requirement Gathering :**- The requirements gathering phase in the SDLC involves collecting and analyzing information from stakeholders to define the functionality, features, and constraints of the software to be developed. This phase aims to understand user needs, business objectives, and system requirements, resulting in a detailed document that serves as a foundation for subsequent stages of the software development process.

2. **Analysis**: - In this phase, the project team gathers and analyzes requirements from stakeholders. This involves understanding the needs and expectations of end-users, defining system functionality, and documenting specifications. The outcome is a comprehensive requirements document that serves as a blueprint for the development process.

3**. Designing**:- The design phase involves creating a detailed blueprint of the system based on the requirements gathered. This includes designing the overall system architecture, specifying data structures, defining algorithms, and creating prototypes or mock-ups. The goal is to provide a clear guide for developers to implement the system.

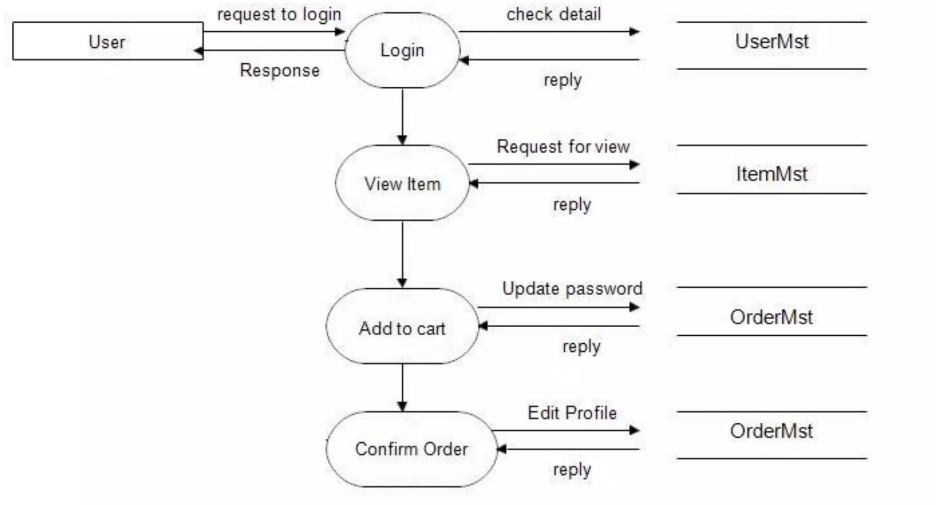
4. **Implementation**:- In this phase, the actual code for the software is written. Developers follow the design specifications and use programming languages to translate the design into executable code. It is a crucial phase where the software product begins to take shape.

5. **Testing** :- The testing phase is dedicated to identifying and fixing defects in the software. It includes various testing levels such as unit testing, integration testing, system testing, and acceptance testing. Testing ensures that the software meets the specified requirements and functions as intended.

6. **Maintenance:**- The maintenance phase involves ongoing support and updates for the software. It includes fixing any post-deployment issues, addressing user feedback, and implementing changes or enhancements to keep the software current and aligned with evolving requirements.

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

Ans. DFD stands for Data Flow Diagram. It is a graphical representation that depicts how data moves through a system. In a DFD, processes, data stores, data flow, and external entities are represented using symbols and arrows to show the flow of data between them. It provides a clear and concise way to visualize the flow of information within a system, making it a valuable tool in systems analysis and design.



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

Ans. ALGORITHM: An algorithm is a step procedure to solve a given problem.

Algorithm of add two numbers

Start

Declare variables n1, n2.

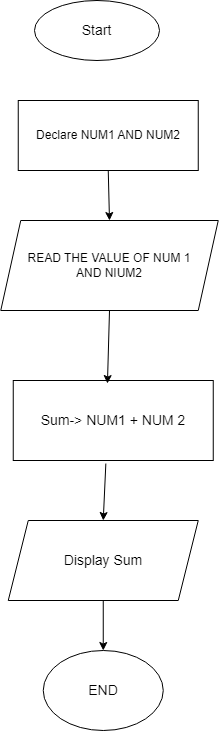
Read the value for n1 and n2.

Sum->n1+n2

Display sum

stop

Flowchart: The Flowchart is the most widely used graphical representation of an algorithm and procedural design workflows. It uses various symbols to show the operations and decisions to be followed in a program. It flows in sequential order.



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

Ans. Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors.

The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

