**Software Engineering Assignment**

**MODULE: 1**

**SE – Overview of IT Industry**

1. What is software? What is software engineering?

Ans:- **Software**

Software refers to the programs, data, and instructions that enable computers to perform various tasks or operations. It encompasses everything from operating systems and applications to games and utilities. In essence, software is the intangible component of a computer system that enables it to function according to predefined instructions.

**Software engineering**

Software engineering, on the other hand, is the discipline concerned with the systematic approach to the design, development, maintenance, and evolution of software systems. It involves applying engineering principles and practices to software development to ensure that the resulting software is reliable, efficient, maintainable, and meets the requirements of its users. Software engineering encompasses various methodologies, techniques, tools, and processes aimed at managing the complexities of large-scale software development projects.

1. **Explain types of software**

Ans : Software can be categorized into several types based on various criteria such as its purpose, functionality, and licensing model. Here are some common types of software:

1. **System Software:** This type of software is essential for the operation of computer hardware and provides a platform for running application software. Examples include operating systems like Windows, macOS, and Linux, as well as device drivers, firmware, and utility programs like antivirus software and disk management tools.

2. **Application Software:** Application software is designed to perform specific tasks or provide specific functionality for end-users. It can range from productivity tools like word processors, spreadsheets, and presentation software to multimedia applications, graphics design programs, and web browsers.

3. **Programming Software:** Programming software consists of tools and applications used by software developers to create, debug, and maintain other software. This includes integrated development environments (IDEs), text editors, compilers, debuggers, and version control systems.

4. **Middleware:** Middleware acts as an intermediary between different software applications, enabling communication and data exchange between them. Examples include web servers, database management systems, and enterprise application integration (EAI) software.

5. **Embedded Software:** Embedded software is specialized software that is built into hardware devices to control their functions. It is commonly found in consumer electronics, automotive systems, industrial machines, and IoT devices.

6. **Enterprise Software:** Enterprise software is designed to support large organizations or businesses in managing their operations and processes. This includes enterprise resource planning (ERP) systems, customer relationship management (CRM) software, and supply chain management (SCM) solutions.

7. **Proprietary Software:** Proprietary software is owned and controlled by a single company or entity, and its source code is typically not available to the public. Users must usually purchase a license to use proprietary software. Examples include Microsoft Office, Adobe Photoshop, and Oracle Database.

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

Ans: SDLC stands for Software Development Life Cycle. It is a structured approach to software development that outlines the stages and activities involved in building software systems. The SDLC consists of several phases, each with its own set of goals, deliverables, and activities.

1. Planning:

- Goal: Define the scope, objectives, and requirements of the project.

- Activities: Conduct feasibility studies, define project goals, gather requirements, establish a project timeline and budget, and assemble the project team.

- Deliverables: Project plan, feasibility report, requirements document, project schedule, and budget.

2. Analysis:

- Goal: Understand and document the detailed requirements of the system.

- Activities: Analyze gathered requirements, clarify ambiguities, prioritize requirements, and create functional and non-functional specifications.

- Deliverables: Requirement specification document, use cases, system requirements specification.

3. Design:

- Goal: Create a blueprint for the software system based on the requirements.

- Activities: Design the system architecture, user interface, database schema, and detailed technical specifications. Identify and document system interfaces and data flows.

- Deliverables: System architecture design, user interface design, database design, detailed design documents.

4. Implementation (Coding):

- Goal: Translate the design into actual code.

- Activities: Write, test, and debug the source code according to the design specifications. This phase involves programming, unit testing, and integration testing.

- Deliverables: Executable code, unit test cases, integration test cases.

5. Testing:

- Goal: Verify that the software meets the specified requirements and is free of defects.

- Activities: Execute test cases to uncover defects, report and track defects, and retest after fixes are made. Testing includes functional testing, performance testing, security testing, and usability testing.

- Deliverables: Test plans, test cases, defect reports, test results.

6. Maintenance:

- Goal: Ensure the ongoing usability and reliability of the software.

- Activities: Address user feedback, fix defects, enhance functionality, and adapt to changes in the operating environment. Maintenance activities may include corrective maintenance, adaptive maintenance, and perfective maintenance.

- Deliverables: Maintenance reports, updated documentation, patches, and new releases.

1. **What is DFD? Create a DFD diagram on Flipkart**

Ans: A Data Flow Diagram (DFD) is a graphical representation of the flow of data within a system. It illustrates how data is input, processed, stored, and output in a system. DFDs are commonly used in the analysis and design of information systems to model the system's functionality at different levels of abstraction.

Creating a DFD for Flipkart, an e-commerce platform, involves identifying the main processes and data flows within the system. Here's a high-level DFD for Flipkart:

Flipkart

Customer

Seller

Catalog Management

Inventory Management

Order System

Cart System

Payment Gateway

Explanation of the DFD elements:

**Customer and Seller**: These represent the external entities interacting with the Flipkart system. Customers browse products, place orders, and make payments. Sellers manage their product listings, inventory, and receive orders.

**Catalog Management**: This process involves managing the product catalog, including adding new products, updating existing products, and removing discontinued products.

**Inventory Management**: This process handles inventory-related tasks, such as updating stock levels, tracking product availability, and managing warehouse logistics.

**Order System**: This process manages the order lifecycle, including order placement, order processing, order fulfillment, and order tracking.

**Cart System**: This process manages the shopping cart functionality, allowing customers to add/remove items, view cart contents, and proceed to checkout.

**Payment Gateway**: This external entity facilitates secure online payments, handling payment processing, transaction authorization, and funds transfer between customers and sellers.

**Data Flows**: Arrows represent the flow of data between processes, entities, and data stores. For example, data flows from the Catalog Management process to the Order System process when a customer selects a product to purchase.

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

Ans:- A flowchart is a graphical representation of a process or algorithm, using various shapes and arrows to illustrate the steps involved and the flow of control. It's commonly used in software development, business processes, and other fields to visually communicate the steps of a procedure or workflow

Here's a simple flowchart to illustrate the process of adding two numbers:

**Input Number 1, Number 2**

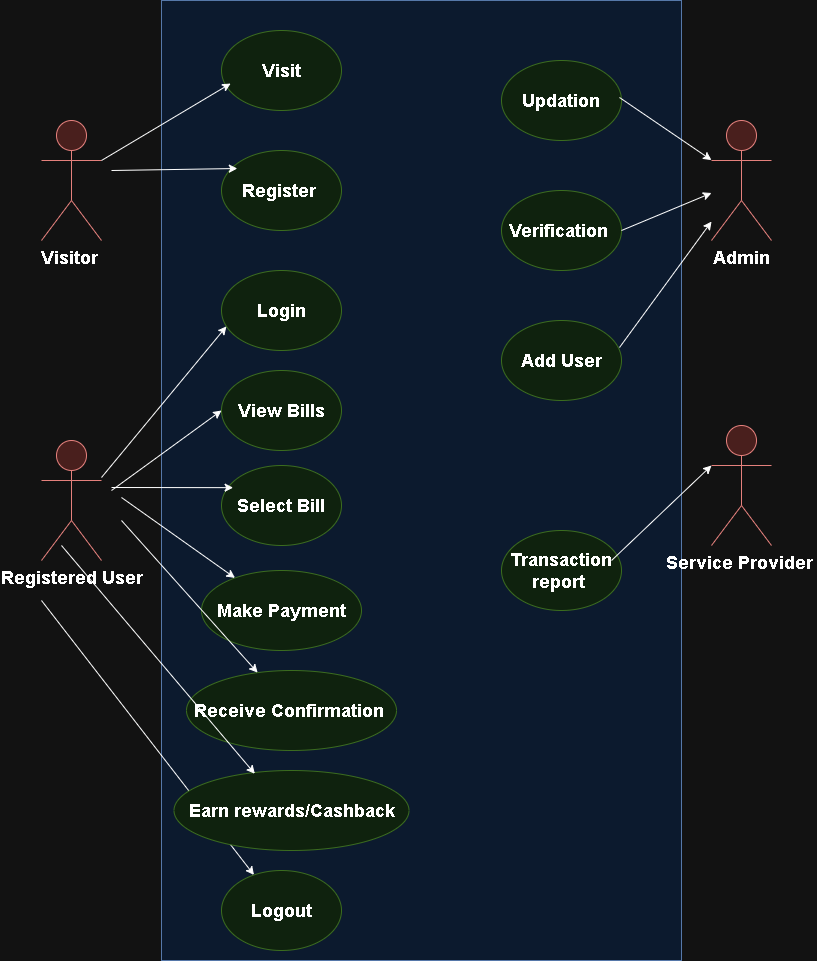
**Sum=Number 1 +Number 2**

**Print Sum**

1. **What is Use case Diagram? Create a use-case on bill payment on paytm.**

Ans: - A Use Case Diagram is a visual representation of the functional requirements of a system from the perspective of its users. It illustrates the interactions between users (actors) and the system to achieve specific goals or tasks.

**USE-CASE OF PAYTM**



**Use-Case for payment on Paytm**