overview of it idustry

1. What is software? What is software engineering?

Software refers to a collection of data or computer instructions that tell the computer how to work. In contrast to hardware, which is the physical component of a computer.

Software Engineering is the systematic application of engineering approaches to the development of software. It encompasses a broad range of activities and methodologies, including:

1.

2.Explain types of software

there is 5 type of software

1. **System Software**

System software provides the basic functions for computer usage and helps manage the computer hardware.

2.Application Software

Application software consists of programs designed to perform specific tasks for users. These tasks can be productive, creative, or recreational.

3. **Middleware**

Middleware is software that connects different applications or services, enabling them to communicate and share data

4. **Programming Software**

Programming software includes tools that developers use to create, debug, maintain, or otherwise support other programs and applications.

5. **Embedded Software**

Embedded software is designed to operate hardware systems and perform specific tasks. It is often found in devices other than computers

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

The **Software Development Life Cycle (SDLC)** is a structured process that guides the development of software applications. It encompasses a series of phases, each with specific tasks and deliverables, to ensure the systematic creation and maintenance of high-quality software. The SDLC aims to produce software that meets or exceeds customer expectations,

Objective: Define the project scope, objectives, and feasibility.

Tasks:

- Identify the project's purpose and goals.

Deliverables:

- Project Charter

- **Requirements Analysis**

  Objective: Gather and document detailed software requirements.

  Tasks:

  - Conduct interviews, surveys, and workshops with stakeholders to gather requirements.

  Deliverables:

  - Requirements Specification Document (RSD)

- **Design**

Objective: Create the architecture and detailed design for the software solution.

Tasks:

  - Develop high-level design (HLD) to define the system architecture.

Deliverables:

  - High-Level Design Document

- **Implementation (or Coding)**

  Objective: Translate design documents into functional software through coding.

  Tasks:

  - Write code according to the design specifications.

  Deliverables:

  - Source Code

- **Testing**

  Objective: Validate that the software works as intended and meets requirements.

  Tasks:

  - Develop test plans, test cases, and test scripts.
  - Perform various testing types: unit testing, integration testing, system testing, and user acceptance testing (UAT).

  Deliverables:

  - Test Plan
  - Test Cases and Test Scripts

- **Deployment**

  Objective: Release the software to the production environment.

  Tasks:

  - Prepare deployment plan, including rollback strategies.
  - Set up the production environment

  Deliverables:

  - Deployment Plan
  - Deployed Application

- **Maintenance**

  Objective: Provide ongoing support and improvements for the software.

  Tasks:

  - Monitor the software for issues and performance.
  - Fix bugs and address defects.

  Deliverables:

  - Maintenance Plan

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

A **Data Flow Diagram (DFD)** is a graphical representation of the flow of data through a system. It illustrates how data is processed by a system in terms of inputs and outputs. DFDs are used to visualize the data processing steps and identify data inputs, outputs, storage points, and data pathways.

```
    +-------------+
    |             |
    |  Customer    |
    |              |
    +--------------+
          |
          | Product Search, Order Details, Payment Informat
 ion
          v
 +--------------------+
 |                    |
 |  Flipkart System   |
 |                    |
 +--------------------+
          |
          | Search Results, Order Confirmation, Payment Con
 firmation
          v
    +--------------+
    |              |
    |  Customer    |
    |              |
    +--------------+
          |
          | Order Request
          v
    +--------------+
    |              |
    |  Supplier    |
    |              |
    +--------------+
          |
          | Order Fulfillment
          v
 +--------------------+
 |                    |
 |  Flipkart System   |
 |                    |
 +--------------------+
          |
          | Payment Request
          v
    +--------------------+
    |                    |
    |  Payment Gateway   |
    |                    |
    +--------------------+
          |
          | Payment Confirmation
          v
 +--------------------+
 |                    |
 |  Flipkart System   |
 |                    |
 +--------------------+
```

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

A **flowchart** is a graphical representation of a process or a system that shows the sequence of steps or actions involved. It uses standardized symbols to illustrate the flow of operations and is widely used in various fields for planning, documenting, and analyzing processes.

```
  +---------+
  |  Start  |
  +---------+
       |
       v
+------------------+
| Input Number 1   |
+------------------+
       |
       v
+------------------+
| Input Number 2   |
+------------------+
       |
       v
+------------------------+
| Add Number 1 and 2     |
+------------------------+
       |
       v
+------------------------+
| Output Result          |
+------------------------+
       |
       v
  +---------+
  |   End   |
  +---------+
```

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

A **Use Case Diagram** is a visual representation of the interactions between users (actors) and a system to achieve a specific goal. It describes the functional requirements of a system and captures the system's behavior from the user's perspective. Use case diagrams are part of the Unified Modeling Language (UML) and are widely used in software engineering to:

Use Case Diagram for Bill Payment on Paytm

Actors:

1. **User**: The person using Paytm to make a bill payment.
2. **Paytm System**: The system facilitating the bill payment process.

3. **Bank System**: External system used for processing the payment.

Use Cases:

1. **Login**: User logs into their Paytm account.
2. **Select Bill Payment**: User selects the bill payment option.
3. **Enter Bill Details**: User enters the bill details (e.g., biller, amount).
4. **Confirm Payment**: User reviews and confirms the payment details.
5. **Make Payment**: Paytm system processes the payment through the bank system.
6. **Receive Confirmation**: User receives confirmation of the payment.

```
│        Paytm System       │
│                       │
│ (1) Login                │
│ (2) Select Bill Payment     │
│ (3) Enter Bill Details      │
│ (4) Confirm Payment         │
│ (5) Make Payment            │
│ (6) Receive Confirmation    │


│ User   │   │Bank System│
```