**Module-1**

1. **What is SDLC?**

* Software development lifecycle 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.

1. **What is software testing?**

Software testing isthe process of evaluating and verifying that a software product or application does what it's supposed to do*.*

1. **What is SRS**

* Software requirement specification (SRS) includes information about all the functional and non-functional requirements for a given piece of software.
* The SRS serves as the main point of reference for the software development team who’ll build the software product, as well as for all other involved stakeholders.

1. **Write SDLC phases with basic introduction**

* SDLC is a structure imposed on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment, and ongoing maintenance and support.
* There are a number of different development models.
* A Software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software.

1. Requirement Gathering
2. Analysis
3. Designing
4. Implementation
5. Testing
6. Maintenance
7. **Requirement Gathering**

* Although requirements may be documented in written form, they may be incomplete, unambiguous, or even incorrect.
* Requirements will Change! Inadequately captured or expressed in the first place
* User and business needs change during the project Validation is needed throughout the software lifecycle, not only when the “final system” is delivered.
* Build constant feedback into the project plan

**There are three problems of the Requirement gathering:**

1. **Lack of Clarity:** It is hard to write documents that are both precise and easy-to-read.
2. **Requirements confusion:** Functional and Non-functional requirements tend to be intertwined.
3. **Requirements Amalgamation:** Several different requirements may be expressed together.

**There are two types of the Requirement:**

1. **Functional Requirement:** describe system services or functions.
2. **Non-Functional Requirement:** are constraints on the system or the development process.

Non-functional requirements may be more critical than functional requirements.

1. **Analysis**

* The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished.
* This phase defines the problem that the customer is trying to solve.
* The deliverable result at the end of this phase is a requirement document.
* This phase starts with the requirement document delivered by the requirement phase and maps the requirements into architecture.
* The architecture defines the components, their interfaces and behaviours.
* Details on computer programming languages and environments, machines, packages, application architecture, distributed architecture layering, memory size, platform, algorithms, data structures, global type definitions, interfaces, and many other engineering details are established

1. **Designing**

* Design Architecture
* Document Implementation
* Plan Critical
* Priority Analysis
* Performance Analysis
* Test Plan
* The Design team can now expand upon the information established in the requirement document.
* The architecture team also converts the typical scenarios into a test plan.

1. **Implementation**

* The implementation phase deals with issues of quality, performance, baselines, libraries, and debugging.
* The end deliverable is the product itself. There are already many established techniques associated with implementation.

1. **Testing**

* Many companies have not learned that quality is important and deliver more claimed functionality but at a lower quality level
* Regression Testing
* Internal Testing
* Unit Testing
* Application Testing
* Stress Testing
* The testing phase is a separate phase which is performed by a different team after the implementation is completed.
* An attitude change must take place to guarantee quality.
* Regardless if testing is done after the-fact or continuously, testing is usually based on a regression technique split into several major focuses, namely internal, unit, application, and stress.

1. **Maintenance**

* Software maintenance is one of the activities in software engineering, and is the process of enhancing and optimizing deployed software (software release), as well as fixing defects.
* configuration and version management
* updating all analysis, design and user documentation

1. **What is OOPs?**

OOPS is a programming paradigm based on the concept of objects, which can contain of data and code, data in the form of fields and code is the form of procedures.

1. **Write Basic Concepts of OOPs**

* Object
* Class
* Encapsulation
* Abstraction
* Inheritance
* Polymorphism

1. **What is object**

It is represent entity of class. It is instance of the class.

1. **What is class**

It is a blueprint of the object and collection of the object.

1. **What is encapsulation**

Encapsulation is refer to integrating code and method into single entity. Encapsulation, class variable are hidden field into the other class and can only be access for the method which they can found. Encapsulation is object oriented procedure, combining of the data and method inside a user defined class.

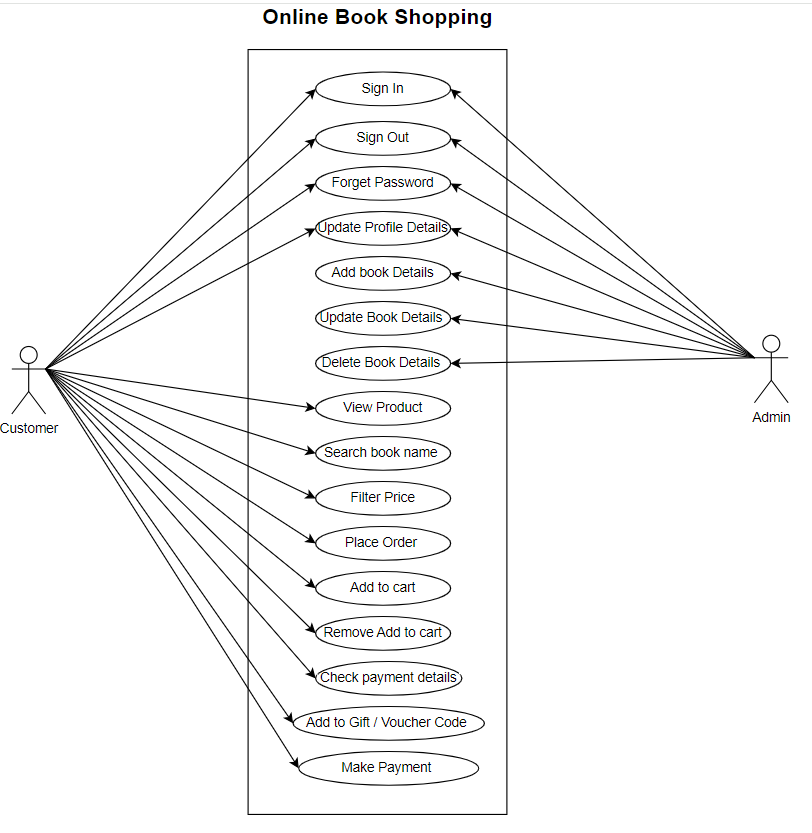
1. **What is inheritance**

Inheritance is ability to adapt behavior of parent class into child class.

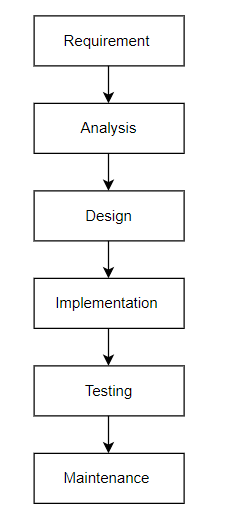
1. **What is polymorphism**

Polymorphism is a single class into a different form.

1. **Draw use case of Online book shopping**

****

1. **Explain Phases of the waterfall model**

****