## Module-1(Fundamental)

1. **What is SDLC?**

Ans. - 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 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. **What is software testing?**

Ans. – Software testing is a process that used to identify the correctness, completeness and quality of developed computer software.

1. **What is Agile methodology**?

Ans. – Agile SDLC model is a combination of iterative and incremental process model with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These build are provided in iterations. Each iteration typically last from about one to three weeks . At the end of the iterations a working product is displayed to the customer and important stakeholders.

1. **What is SRS?**

Ans. – A software requirement specification is a complete description of the behavior of the system to be developed. It includes a set of use cases that described all of the interaction that the users will have with the software. Use cases are also known as functional requirements. It also include Non- functional requirements. SRS are recommended by IEEE-830-1998. This standard describes possible structure, desirable content, and qualities of a software requirement specification.

1. **What is oops?**

Ans.- An object –based programming language is one which easily supports object-orientation. Its identifying objects and assigning responsibilities to these objects. Objects communicate to other objects by sending message. Messages are received by the methods of an object. Object is like a black box where internal details are hidden.

1. **Write basic concept of oops.**

Ans. – Following are the basic concept of oops –

* Object
* Class
* Encapsulation
* Inheritance
* Abstraction
* Polymorphism
* Overriding
* Overloading

1. **What is object?**

Ans. –An object represent individual, identifiable item, unit, or entity, either real or abstract, with a well-defined role in the problem domain. An object is anything to which a concept applies. This is the basic unit of object oriented programming. That is both data and function that operate on data are bundled as a unit called as object.

1. **What is class?**

Ans. - Class is a structure in which we can have member function and member variables. A class represent an abstraction of the object and abstracts the properties and behavior of that object. Class can be considered as the blueprint or a template for an object and describes the properties and behavior of that object, but without any actual existence.

1. **What is encapsulation?**

Ans. – Encapsulation is the practice of including in an object everything it needs hidden from other objects. The internal state is usually not accessible by other object. Encapsulation in Java is the process of wrapping up of data and behavior of an object into a single unit; and unit here is a Class.

1. **What is inheritance?**

Ans.- Inheritance means that one class inherits the characteristics of another class. This is also called a “is a “ relationship. Inheritance describes the relationship between two classes. A class can get some of its characteristics from a parent class and then add unique features of its own. This features help to reduce the code size.

1. **What is Polymorphism?**

Ans.- Polymorphism means “having many forms”. It allows different objects to respond to the same message in different ways, the response specific to the type of the object. The ability to change form is known as polymorphism. There is two type of polymorphism in Java-

1. compile time polymorphism (overloading)
2. Runtime polymorphism (overriding)
3. **Draw use case on online book shopping.**

<https://drive.google.com/file/d/1IVyjP7EnsBLHbBea8Ob-2N1JSD_jvTCB/view?usp=drive_link>

1. **Draw use case on online bill payment system.**

<https://drive.google.com/file/d/1LXbz08P2NW0vnxdWRAYP8x3xgvwALOlw/view?usp=sharing>

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

Ans.- The Software Development Life Cycle is a process used by software development team to design, develop, test, and deploy quality software. It typically consists of several phases:

1. Requirement Gathering – In this phase, projects goals, requirements, timelines, and resources are defined. This set the foundation for the entire development process. Requirements definitions usually consists of natural language, supplemented by diagrams and tables.
2. Analysis Phase- The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished. This phase helps in identifying the scope and functionality of the software.
3. Design Phase- During this phase, the software architecture, system specifications, and user interface design are created based on the requirements gathered. This phase serves as a blue print for the development team.
4. Implementation Phase- Also known as the coding phase, this is where developers write and implement the code according to the design specifications. It involves actual programming and may also include integration of third- party components.
5. Testing Phase- The software is thoroughly tested to ensure it meets the specified requirements and functions correctly. This include unit testing, integration testing, system testing, user acceptance testing.
6. Maintenance Phase- 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.
7. **Explain phases of waterfall model.**

Ans.- The waterfall model typically consists of the following phases:

1. Requirements Gathering : This phase involves gathering and documenting requirements from stakeholder.
2. Analysis :The analysis phase defines the requirements of the system, independent how these requirements will be accomplished.
3. Design : In this phase, the system architecture and design are developed based on the requirements gathered.
4. Implementation : The actual coding or development of the system take place in this phase.
5. Testing : Once the implementation is complete, the system is tested to ensure it meets the specified requirements.
6. Maintenance : This phase involve ongoing maintenance and support of the system, including bug fixes and updates.
7. **Write phases of spiral model.**

Ans.-The spiral model typically consists of the following phases:

1. Planning : This phase involves defining the project objectives, constraints, and alternatives. Risks are identified, and strategies to mitigate them are developed.
2. Risk analysis : In this phase, the identified risks are analyzed in detail. This involves assessing the impact of risks on the project and determining how to address them.
3. Engineering : During this phase, the project is developed incrementally. Each iteration of development includes requirements gathering, design, coding, and testing.
4. Evaluation: After each iteration, the project is evaluated to assess its progress and determine whether it meets the project objectives. This evaluation helps in deciding whether to proceed to the next iteration or to make adjustments.
5. **Explain working methodology of agile model and also write pros and cons**.

Ans.-The Agile methodology is a software development approach that emphasizes flexibility, collaboration, and iterative progress. Here’s a breakdown of its working methodology and pros and cons:

Working methodology:

1. Iterative development : Agile breaks the projects into small increments called iterations or sprints, typically 2-4 weeks long. Each iteration results in a potentially shippable product increment.
2. Cross-functional team: Agile teams are self-organizing and cross- functional, comprising members with different skills (developers, testers, designers) who collaborate closely throughout the project.
3. Customer collaboration: Continuous feedback from stakeholders, including customers, is integral to Agile. This feedback loop ensures the product meets evolving needs and requirements.
4. Continuous improvement: Regular retrospectives enable teams to reflect on their processes and outcome, identifying areas for improvement and making adjustments accordingly.

Pros:

* It is very realistic approach to software development.
* Promote teamwork and cross training.
* Functionality can be developed rapidly and demonstrated.
* Resource requirements are minimum and easy to manage.
* Delivers easy partial working solutions.

Cons:

* Not suitable for handling complex dependency.
* More risks of sustainability, maintainability, and extensibility.
* An overall plan, an agile leader and agile product manager practice is must without which it will not work.
* Transitioning to agile may face resistance from team members accustomed to traditional, more linear development methodologies.
* There is minimum documentation generated.
* Transfer of technology to new member may be quite challenging due to lack documentation.

1. **Draw use case on online shopping product using COD.**

Ans.-<https://drive.google.com/file/d/16pTAMlXwqh1lij8msUQ47Q5WdidKMApq/view?usp=sharing>

1. **Draw use case on online shopping product using payment gateway.**

Ans.-<https://drive.google.com/file/d/1UqQKMMeAqQ68pBsux8_DhXecL2zZ-3fg/view?usp=sharing>