Module –1 (Fundamental)

1) What is SDLC?

- 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.

2) What is software testing?

- Software testing is a process used to identify the correctness, completeness, and quality of developed computer software. It involves executing the software/system to find errors, gaps, or missing requirements compared to the actual requirements.

3) What is agile methodology?

- Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.

4) What is SRS?

- A software requirements specification (SRS) is a complete description of the behavior of the system to be developed.

5) What is oops?

- Object-Oriented Programming (OOP) is a programming paradigm that organizes software design around objects, which are instances of classes.

6) Write Basic Concepts of oops

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

7) What is object

- An object is an instance of a class that contains data in the form of attributes and functions in the form of methods. It represents a real-world entity with specific characteristics and behaviors.

8) What is class

- Class is a group of different types of variable and function.

9) What is encapsulation

- To bind a data and code in to a single unit is called encapsulation.

10) What is inheritance

- The object of one class can Acquire the property of object of another class is called inheritance
- Creating a new class from an existing class is called an inheritance.

11) What is polymorphism

- One name multiple form it means polymorphism.
  Two types of polymorphism

- **Method overloading**

When there is one method in a single class having the same name but with the different number of argument and that data types it is called method overloading.

- **Method overriding**

When there is a same method prototype in your both base class and derived class and if you called that method using that object derived class than only derived class method will be called you can say that method of derived class overrides the method of base class.

12) Write SDLC phases with basic introduction

- **Requirement Gathering**

  Understand what the client or user needs.

- **Analysis**

  Model and specify the requirements

- **Design**

  Plan the software's architecture.

- **Implementation**

  Developers write the actual code based on the design.

- **Testing**

  Test the software for bugs, errors, or issues.

- **Maintenance**

  Repair defects and adapt the solution to the new requirements

13) Explain Phases of the waterfall model

- The Waterfall Model is a step-by-step software development process where each phase flows into the next — just like a waterfall.

- **Requirement Gathering and Analysis**

- Collect all requirements from the client or user.
- Create a clear document (SRS – Software Requirement Specification).

- **System Design**

- Plan how the software will be built.
- Includes architecture, technologies, database design, and interfaces.

- **Implementation (Coding)**

- Developers write the actual code based on the design documents.
- Each module is developed and tested individually

- **Integration and Testing**

- Combine all modules and test the system as a whole.
- Identify and fix bugs.

- **Deployment**

- Deliver the software to the client or deploy it in the production environment.
- Users start using the final product.

- **Maintenance**

- Fix issues that come up after deployment.
- Fix bugs, add new features, or update the system based on user feedback.

14) Write phases of spiral model

- **Planning Phase**

  Determination of objectives alternatives and constraints

- **Risk Analysis Phase**

  Analysis of alternatives and identification/ resolution of risk

- **Engineering Phase**

  Assessment of the result of engineering

- **Evaluation Phase**

  Development of the next level product

15) Write agile manifesto principles

- Customer satisfaction
- Welcome changing requirements
- Frequent delivery
- Face-to-face communication
- Support and trust
- Working software is the primary measure of progress
- Sustainable development
- Technical excellence and good design
- Simplicity
- Self-organizing teams
- Regular reflection and improvement

16) Explain working methodology of agile model and also write pros and cons

➢ **Product Backlog**

- What is it: The product backlog is a list of all desired features**,** enhancements, fixes, and tasks required for the software project.

- **Sprint Planning**

- What is it**:** Before starting each sprint, the team meets to decide what work will be completed in the upcoming sprint?

- **Sprint Execution**

- What is it**:** This is where the actual development work happens

- **Daily Stand-ups (Daily Scrum)**

- What is it: A brief, daily meeting (usually 15 minutes) where team members share their progress.

- **Sprint Review**

- What is it**:** At the end of the sprint, a Sprint Review meeting is held to demonstrate the completed work.

- **Sprint Retrospective**

- What is it**:** After the sprint review, the team holds **a** Sprint Retrospective to reflect on the sprint.

- **Repeat the Process**

- Next Sprint**:** The process starts again with Sprint Planning for the next iteration.

Pros

- Is a very realistic approach to software development Promotes teamwork and cross training
- Suitable for fixed or changing requirements Delivers early partial working solutions.
- Enables concurrent development and delivery within an overall
- planned context.
- Little or no planning required Easy to manage
- Gives flexibility to developers

Cons

- Not suitable for handling complex dependencies.
- More risk of sustainability, maintainability and extensibility.
- An overall plan, an agile leader and agile PM practice is a must without which it will not work.
- to be delivered, and adjustments to meet the deadlines.
- There is very high individual dependency, since there is minimum documentation generated.