

Module - 1

1) What is SDLC?

- SDLC stands for Software Development Life Cycle. It's a structured process that helps plan, design, build, test and maintain software. The SDLC is a cost-effective and time-efficient way to create high quality software.

2) What is software testing?

- Software testing is the process of evaluating software to ensure it meets requirements and functions as expected.

3) What is agile methodology?

- Agile methodology is a project management approach that emphasizes iterative development, flexibility and collaboration to adapt to changing requirements.

4) What is SRS?

- SRS stands for Software Requirements Specification. It's a detailed document outlining the functional and non-functional requirements of a software system, serving as a contract between the client and developers and acting as a guide for development, testing and validation.

5) What is OOPS?

- OOPS stands for object oriented programming system. OOPS is a

programming paradigm that organizes software design around data or objects rather than functions and logic.

6) Write basic concepts of oops.

- Oops include encapsulation, inheritance, polymorphism and abstraction.
- **Class** : Class is a group of different types of variables and functions.
- **Object** : An object is a specific instance of a class.
- **Encapsulation** : To bind a data and code into a single unit it's called Encapsulation.
- **Inheritance** : The object of class can acquire the properties of object of another class it's called Inheritance.
- **Polymorphism** : one name multiple form.

(I) Compile time polymorphism : When there is more than method in a single class having a same name but different numbers of arguments and their data types when it's called method overloading.

(II) Runtime polymorphism : when there is same method prototype in your both base class and derived class and if you call that method using the object of derived class then only derived class method will be called. you can say that method of derived class overrides the method of base class.

7) What is object?

- In oops an object is a fundamental unit that encapsulates both data and actions that operate on data.
- It's essentially an instance of a class, acting as a blueprint for creating

objects. Objects are the building blocks of opp applications.

- They represent real-world entities or concepts, combining data and actions and are created from class blueprints.

8) What is class?

- In opps a class is a blue print for creating objects. It defines the structure and behaviour of objects by specifying their attributes, method and relationship
- A user defined data type that describes a type of object.

9) What is Encapsutulation?

- Encapsulation in opps is the practice of bundling data and method into a single unit called a class.
- This concepts protects data from outside interference and makes easier to maintain.

10) What is inheritance?

- In opps inheritance ia a mechanism where a class inherits properties and behaviours from another class.
- **Types of Inheritance :**

I) single Inheritance

II) Multiple Inheritance

III) Hierarchical Inheritance

IV)Multilevel Inheritance

11) What is polymorphism?

- **Polymorphism** : one name multiple form.

(I) Compile time polymorphism : When there is more than method in a single class having a same name but different numbers of arguments and their data types when it's called method overloading.

(II) Runtime polymorphism : when there is same method prototype in your both base class and derived class and if you call that method using the object of derived class then 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?

- The seven phases of SDLC.

1) Planning : Defining the software's purpose, scope and goals.

2) Requirements Analysis : Gathering and documenting users needs and expectations.

3) design : Creating the software's architecture, user interfaces and database.

4) Coding : Writing the actual software code.

5) Testing : Identifying and fixing bugs to ensure quality.

6) Deployment : Releasing the software to users.

7) Maintenance : Continuously supporting and improving the software

after deployment.

13) Explain phases of waterfall model.

- **Requirements** : This phase involves gathering and documenting all project requirements from stakeholders, defining the functionality, purpose, and specifications of the final product.
- **Design** : Based on the requirements, a comprehensive design is developed, outlining the system architecture, workflow, and overall structure.
- **Implementation** : This phase involves coding and building the software components according to the design specifications.
- **Testing** : Once the implementation is complete, rigorous testing is conducted to ensure the software functions as intended and meets the defined requirements.
- **Deployment** : The tested software is then deployed to the target environment, either in a customer's system or released to the market.
- **Maintenance** : After deployment, the software is maintained to provide ongoing support, bug fixes, and enhancements to ensure its functionality and usability.

14) Write phases of spiral model.

- **Planning** : This phase involves defining the project's objectives, requirements, and constraints. It also includes identifying stakeholders and gathering information about their needs and

expectations.

- **Risk Analysis** : This is a crucial phase where potential risks are identified, assessed, and strategies for mitigating them are developed. This proactive approach helps to reduce the impact of potential issues later in the development process.
- **Product Development** : This phase involves the actual development of the software product, including coding, testing, and integration. It also includes creating prototypes and gathering user feedback.
- **Evaluation** : This phase focuses on evaluating the software product against established requirements, gathering user feedback, and planning for the next iteration. This feedback helps to refine the product and address any issues identified during the evaluation.

15) Write agile manifesto principles.

- Through Agile project management, techniques such as self-management, self-organization and continuous delivery achieve benefits beyond the domain of software development.
- **Principles of agile :**

Individuals and interactions over processes and tools:

Agile emphasizes collaboration, communication, and team dynamics over rigid processes and tools.

Working software over comprehensive documentation:

Delivering functional, working software is prioritized over extensive documentation, with a focus on value and usability.

Customer collaboration over contract negotiation:

Agile encourages close collaboration with stakeholders throughout the project to ensure that the final product meets their needs.

Responding to change over following a plan:

Agile teams are adaptable and embrace change, prioritizing flexibility and responsiveness over rigid plans.

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

- **Iterative and Incremental :**

Agile projects are broken down into smaller iterations or "sprints," with teams delivering working software at the end of each sprint.

- **Continuous Feedback :**

Regular feedback loops are established to gather input from stakeholders and ensure that the product is evolving in the right direction.

- **Cross-functional Teams :**

Agile teams are typically cross-functional, with members possessing a range of skills and responsibilities.

- **Self-Organizing Teams :**

Agile teams are empowered to make decisions and manage their own work, promoting autonomy and accountability.

Agile model Pros and Cons :

Pros

Cons

- | | |
|---------------------------------------------------------------|---------------------------------|
| • Motivation to finish the sprint | Losing the track of the project |
| • The focus on quality | No big picture of the project |
| • All team members have the
access to the project progress | Not clear defining each role |