

Module–1(Fundamental)

1) What is SDLC:-

- Software development life cycle is a process followed for software building within a software organization. SDLC consists of an accurate plan which describes development, maintainance, replacement, and enhancement of specific software. The life cycle defines a method for improving the quality of software and the all-around development process.

2) What is software testing?

- Software Testing is a process used to identify the correctness, completeness, and quality of developed computer software.

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.
- Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing. At the end of the iteration a working product is displayed to the customer and important stakeholders

4) What is SRS?

- A software requirements specification (SRS) is a whole description of the behaviour of the system that is being developed. It includes a set of use cases that describe all of the interactions that the users will have with the software .

5) What is oops?

- Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or [objects](#), rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior.

6) Write Basic Concepts of oops?

- The basic concepts of oops are:-

1. Object
2. Class
3. Encapsulation
4. Inheritance
5. Polymorphism
6. Overriding
7. Overloading
8. Abstraction

7) What is object?

- An object is a fundamental building block of object-oriented programming (OOP) and represents a real-world entity.

8) What is class?

- A class represents an abstraction of the object and abstracts the properties and behavior of that object. Class can be considered as the blueprint or definition or a template for an object and describes the properties and behaviour of that object, but without any actual existence.

9) What is encapsulation?

- 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 objects. Encapsulation in Java is the process of wrapping up data and behavior of an object into a single unit; and the unit here is a Class.

10) What is inheritance?

- Inheritance means that one class inherits the characteristics of another class. This is also called a relationship. One of the most useful aspects of object-oriented programming is code reusability. As the name suggests, Inheritance is the process of forming a new class from an existing class that is from the existing class called as base class, new class is formed called as derived class. This is a very important concept of object-oriented programming since this feature helps to reduce the code size. 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.

11) What is polymorphism?

- 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 most important aspect of an object is its behaviour (the things it can do). A behaviour is initiated by sending a message to the object (usually by calling a method). The ability to use an operator or function in different ways in other words giving different meaning or functions to the operators or functions is called polymorphism.

12) Draw Usecase on Online book shopping?

<https://drive.google.com/file/d/1tdwtVaXWEYkk9oRZb1GGuKrBeH8MQzdR/view?usp=sharing>

13) Draw Usecase on online bill payment system (paytm)?

<https://drive.google.com/file/d/1stvpBVJbFttwv4Y5AIpId-4t4yW-eYLO/view?usp=sharing>

14) Write SDLC phases with basic introduction?

- The Software Development Life Cycle (SDLC) is a seven-phase process that's the foundation of any software development:
 1. Planning
This phase includes tasks like cost-benefit analysis, scheduling, resource estimation, and allocation.
 2. Requirements analysis
 3. Design
In this phase, software engineers analyze requirements and identify the best solutions to create the software. This includes defining the architecture, user interface, security, and programming.
 4. Coding
Also known as implementation, this is when the development team codes the product.
 5. Testing:-
The development team combines automation and manual testing to check the software for bugs.
 6. Deployment
After successful testing, the software is deployed to a production environment and made available to end-users.
 7. Maintenance
This phase includes ongoing support, bug fixes, and updates to the softwa

15) Explain Phases of the waterfall model?

- The Waterfall Model has six phases which are:
 - 1. Requirements:** The first phase involves gathering requirements from stakeholders and analyzing them to understand the scope and objectives of the project.
 - 2. Design:** Once the requirements are understood, the design phase begins. This involves creating a detailed design document that outlines the software architecture, user interface, and system components.
 - 3. Development:** The Development phase include implementation involves coding the software based on the design specifications. This phase also includes unit testing to ensure that each component of the software is working as expected.
 - 4. Testing:** In the testing phase, the software is tested as a whole to ensure that it meets the requirements and is free from defects.

5. Deployment: Once the software has been tested and approved, it is deployed to the production environment.

6. Maintenance: The final phase of the Waterfall Model is maintenance, which involves fixing any issues that arise after the software has been deployed and ensuring that it continues to meet the requirements over time.

16) Write phases of spiral model?

- There are four phases of spiral model:-
 - 1.Planning:-It includes determination of objectives, alternatives and constraints
 - 2.Risk analysis:-It includes analysis, alternatives and identification/resolution of risks.
 - 3.Customer evaluation:-It includes assessment results of engineering
 - 4.Engineering:-It includes development of the quality of product

17) Write agile manifesto principles?

- Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release.
- Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer. Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability

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

- 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. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing. At the end of the iteration a working product is displayed to the customer and important stakeholders.

Pros	Cons
<ul style="list-style-type: none">• Is a very realistic approach to software development.• Promotes teamwork and cross training.• Functionality can be developed rapidly and demonstrated.• Resource requirements are minimum.	<ul style="list-style-type: none">• 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

<ul style="list-style-type: none"> • Suitable for fixed or changing requirements • Delivers early partial working solutions. • Good model for environments that change steadily. • Minimal rules, documentation easily employed. • Enables concurrent development and delivery within an overall planned context. • Little or no planning required. • Easy to manage Gives flexibility to developers 	<p>which it will not work.</p> <ul style="list-style-type: none"> • Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines. • Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction. • There is very high individual dependency, since there is minimum documentation generated. • Transfer of technology to new team members may be quite challenging due to lack of documentation.
---	---

19) Draw usecase on Online shopping product using COD?

<https://drive.google.com/file/d/1Lh-TQ7sg2jxTJWX4h4mK2NtyNi8xnaxi/view?usp=sharing>

20) Draw usecase on Online shopping product using payment gateway?

https://drive.google.com/file/d/1hoaAJ6lFWfvyR-TM_o6pV2MFo65-CgDh/view?usp=sharing