**1. What is a SDLC**

**A**. 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.

The methodology within the SDLC process vary across industries and organizations, but standards such as ISO/IEC 120007 represent processes that establish a lifecycle for software, and provide a mode for the development, acquisition, and configuration of software systems.

**2. What is software testing?**

**A.** Testing is the process of evaluating a system or its component (s) with the intent to find that whether it satisfies the specified requirement or not the specified requirements or not.

In simple words testing is executing a system in order to identify any gaps,or missing requirements in contrary to the actual desire or requirements.

**3. What is agile methodology ?.**

**A**. 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.

Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis design , coding , unit testing and acceptance testing.

**4. What is SRS ?.**

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

Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance requirements, quality standards, or design constraints).

**5. What is oops**

**A.** Identifying objects and assigning responsibilities to these objects.

Objects communicate to other objects by sending messages.

Messages are received by the methods of and object.

Object-oriented programming has a wab of interacting objects, each house-keeping its own state.

**6. Write Basic Concepts of oops**

**A.** An object is like a black box

The internal details are hidden

Object is derived from abstract data type

Objects of a program interact by sending messages to each other.

**7. What is object**

**A.** Tangible Things as a car, printer, ….

Roles as employee, boss, ….

Incidents as flight , overflow, ….

Interactions as contract, sale, ….

An object represents an individual, identifiable item, unit, or entity, either real or abstract, with a well- defined role in to problem domain.

**8. What is class.**

**A.** When you define a class you define a blueprint for an object.

This don’t actually define any data, but it does define what the class name mean , that is, what an object of the class will consist of and what operations can be performed on such an object.

Class can be considered as the blue print or definition or a template for an object and describes the properties and behavior of that object, but without any actual existence.

**9. What is encapsulation.**

**A.** Encapsulation is the practice of including in an object everything it need hidden from other objects. The internal state is usually not accessible by other objects.

Encapsulation is placing the and the functions that work on that data in the same place. While working with procedural languages, it is not always clear which function work on which variables but object-oriented programming provides you framework to place the data and the relevant function together in the same object.

**Encapsulation** in java is the process of wrapping up of data (properties) and behavior

(method) of an object into a single unit; and the unit here is a Class (or interface)

10. What is inheritance.

A. Inheritance mean that one class inherits the characteristics of another class. This is also called a “is 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 called as base class, new class is formed called as derived class.

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 .**

**A.** 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 function to the operators functions is called polymorphism.

**12.Draw Use case on Online book shopping.**

**A.** <https://drive.google.com/file/d/1HIcntttJ-Mxb_jtC4oJG9i8Zi8h9edmS/view?usp=sharing>

**13. Draw Use case on online bill payment system (paytm)**

**A.** <https://drive.google.com/file/d/1p1OLPMMyMYkdaTnmGEojUWtCr-fhPhqY/view?usp=sharing>

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

A. 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

The methodology withing the SDLC process can vary across industries and organizations, but standards such ISO/IEC 12207 represent processes that establish a lifecycle for software , and provide a mode for the development , acquisition, and configuration of systems.

**15. Explain phase of the waterfall model**

**A.** The classical software lifecycle models the software development as a step by-step “water fall” between the various development phases

Requirements collection

Analysis

Design

Implementation

Testing

Maintenance

The waterfall is unrealistic for many reasons especially

Requirements must be “frozen” to early in the life cycle

Requirement are validated too late

**16. Write phase of spiral model.**

**A.** Spiral Model is very widely used in the software industry as it is in synch with the natural development process of any product I.e. with maturity and also involves minimum risk for the customer as well as the development firms. Following are the typical uses of Spiral model development firms. Following are the typical uses of Spiral model.

When costs there are budget constraint and risk evaluation is important .

Long -term project commitment because of potential changes to economic priorities as the requirement change with time

Customer is not sure of their requirements which are usually the case.

**17. Write agile manifesto principles.**

**A.** Customer satisfaction through early and continuous software delivery Customers are happier when the receive working software at regular intervals, rather than waiting extended periods of time between releases

Accommodate changing requirements throughout the development process – The ability to avoid delay when a requirement or feature request changes.

Frequent delivery of working software – Scrum accommodates this principle since the team operates in software sprints or iteration that ensure regular delivery of working software.

Collaboration between the business stakeholders and developers throughout the project – Better decisions are made when the business and technical team are aligned.

**18. Explain Working methodology of agile model and also write pros and cons.**

**A.** 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.

Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design , coding unit testing and acceptance testing.

Pros

* Is a very realistic approach to software development.
* Promotes teamwork and cross training .
* Functionality can be developed rapidly and demonstrated .
* Resource requirements are minimum.
* Little or no planning required.

**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 must without which it will not work.
* Strict delivery management dictates the scope functionality to be delivered, and adjustments to meet the deadlines.
* There is very high individual dependency. Since there is minimum documentation generated.

**19. Draw use case on Online shopping product using COD.**

A. <https://drive.google.com/file/d/1hW7_2RGrNycw9gT-7su3-4JY5woGyV58/view?usp=sharing>

20. Draw use case on Online Shopping product using payment gateway.

A. <https://drive.google.com/file/d/16EPeYQpVXoTWSQI46YXjBAUQvNIbHkKO/view?usp=sharing>