**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.

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 can vary across industries and organizations, but standards such as ISO/IEC 12207 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?**

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

‘The process consisting of all life cycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products to determine that they satisfy speciﬁed requirements, to demonstrate that they are ﬁt for purpose and to detect defects.’

When asked, people often think that Testing only consists of running tests, i.e. executing the software Test execution is only a part of testing, but not all of the testing activities.Test activities exist before and after test execution

Process: Testing is a process rather than a single activity.

All Life Cycle Activities: Testing is a process that’s take place throughout the Software Development Life Cycle (SDLC).

The process of designing tests early in the life cycle can help to prevent defects from being introduced in the code. Sometimes it’s referred as “verifying the test basis via the test design”.

The test basis includes documents such as the requirements and design speciﬁcations.

**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

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.

**4.what is SRS**

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

It includes a set of use cases that describe all of the interactions that the users will have with the software.

Use cases are also known as functional requirements. In addition to use cases, the SRS also contains nonfunctional (or supplementary) requirements.

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

This standard describes possible structures, desirable contents, and qualities of a software requirements speciﬁcation.

**5.what is oops**

**Object oriented programming system**

Identifying objects and assigning

responsibilities to these objects. Objects communicate to other objects by sending messages.

Messages are received by the methods

Of an object An object is like a black box.

The internal details are hidden.

Object is derived from abstract data type

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

Objects of a program interact by sending messages to each

other.

**6.write basic concept of oops**

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

**7.what is object**

An object represents an 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(OOP).

That is both data and function that operate on data are bundled as a unit called as object.

**8.what is class**

When you deﬁne a class, you deﬁne a blueprint for an object.

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

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 behavior of that object, but without any actual existence.

An object is a particular instance of a class which has actual existence and there can be many objects (or instances) for a class.

In the case of a car or laptop, there will be a blueprint or design created ﬁrst and then the actual car or laptop will be built based on that.

We do not actually buy these blueprints but the actual objects.

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

Encapsulation in Java is the process of wrapping up of data (properties) and behavior (methods) of an object into a single unit; and the unit here is a Class (or interface).

Encapsulate in plain English means *to enclose or be enclosed in or as if in a capsule*. In Java, a class is the capsule (or unit).

**10.what is inharitance**

Inheritance means 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 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**

Poly refers to many. That is a single function or an operator functioning in many ways diﬀerent upon the usage is called polymorphism.

E.g. the message *displayDetails()* of the Person class should give diﬀerent results when send to a Student object (e.g. the enrolment number)

**12.draw usecase on online book shopping**

browse home page

advanced search

give rating

maintain account

update shopping cart

update profile

login

place an order

register

log out

check out

log in

manage shopping cart

search book

logout

add or delete member

add or delete cc

manage orders

add or delete books

delete or add catagory

administrator

user

**13.draw use case on online bill payment system**

login page

money transfer

update balance

provide QR

link to all bank

security features

add account

add credit card or debit card details

update transection history

to mobile number

to bank

scan QR

avail offers

admin

user

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

* **requirement analysis**

Establish Customer Needs

* **Analysis**

Model And Specify the requirements- “What”

* **Product design.**

Model And Specify a Solution – “Why”

* **Implementation**

Construct a Solution In Software

* **Testing.**

Validate the solution against the requirements

* **Post-production maintenance.**

Repair defects and adapt the solution to the new requirements

**15.explain phases of the waterfall model**

* ***Requirements****: The first phase involves understanding what needs to design and what is its function, purpose, etc. Here, the specifications of the input and output or the final product are studied and marked.*
* ***System Design****: The requirement specifications from the first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The software code to be written in the next stage is created now*.
* ***Implementation****: With inputs from system design, the system is first developed in small programs called units, which are integrated into the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.*
* ***Integration and Testing****: All the units developed in the implementation phase are integrated into a system after testing of each unit. The software designed, needs to go through constant TESTING to find out if there are any flaws or errors Testing is done so that the client does not face any problem during the installation of the software*.
* ***Deployment of System****: Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.*
* ***Maintenance****: This step occurs after installation, and involves making modifications to the system or an individual component to alter attributes or improve performance. These modifications arise either due to change requests initiated by the customer, or defects uncovered during live use of the system. The client is provided with regular maintenance and support for the developed software*.

**16.write phases of spiral model**

t has four stages or phases: The planning of objectives, risk analysis, engineering or development, and finally review. A project passes through all these stages repeatedly and the phases are known as a Spiral in the model.

1. **Determine objectives and find alternate solutions –** This phase includes requirement gathering and analysis. Based on the requirements, objectives are defined and different alternate solutions are proposed.
2. **Risk Analysis and resolving –** In this quadrant, all the proposed solutions are analyzed and any potential risk is identified, analyzed, and resolved.
3. **Develop and test:** This phase includes the actual implementation of the different features. All the implemented features are then verified with thorough testing.
4. **Review and planning of the next phase –** In this phase,the software is evaluated by the customer. It also includes risk identification and monitoring like cost overrun or schedule slippage and after that planning of the next phase is started.

**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**

Is a very realistic approach to software development Promotes teamwork and cross training.

Functionality can be developed rapidly and demonstrated. Resource requirements are minimum.

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

**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.

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**

VIEW ITEM

REGISTRATION

L0G IN

ADD CATAGORY

ADD ITEM

MANAGE ORDER

ADD PAYMENT METHOD

MAKE ORDER

ADMIN

USER

**20.draw usecase on online shopping product using payment gateway.**