* **What is SDLC ?**

Ans - SDLC mean software development lifcycle ,

SDLC is a structure imposed on the development of a software product that define the process for planning , implementation,testing, documantation,deploymentand on going maintenance and support,there are a number of different devlopment models

* A software development life cycle is essentially a series of step or phases, that providea model for the development and life cycle management of an application or piece of software
* The methodology within the sdlc process can vary across industries and organization, but standard such as ISO/IEC 12207 represent process that establish a life cycle for software, and provide a mode for development , acquisition,and configuration software system

* **What is software testing ?**

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

* **What is agile methodology ?**

Ans - 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

* **What is SRS ?**

ANS - Software Requirement Specification 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). Recommended approaches for the specification of software requirements are described by IEEE 830-1998. This standard describes possible structures, desirable contents, and qualities of a software requirements specification.

* **What is oops ?**

Ans -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.

* **What is object ?**

Ans- 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.

* **What is class ?**

Ans - 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.

* **What is encapsulation ?**

Ans -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 objectoriented 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).

* **What is inheritance ?**

Ans - 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.

* **What is polymorphism ?**

Ans - 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).

The ability to change form is known as polymorphism. There is two types of polymorphism in Java Compile time polymorphism(Overloading) Runtime polymorphism(Overriding)

* **Write SDLC phases with basic introduction ?**

Ans -

| Requirement /  Collection gathering | Establish Customer Needs |
| --- | --- |
| Analysis | Model And Specify the requirements- “What” |
| Design | Model And Specify a Solution – “Why” |
| Implementation | Construct a Solution In Software |
| Testing | Validate the solution against the requirements |
| Maintenance | Repair defects and adapt the solution to the new requirements |

* Analysis Phase
* Design Phase
* Implementation Phase
* Testing Phase
* Maintenance Phase
* **Explain phases of waterfall model**

Ans - The waterfall is unrealistic for many reasons, especially: Requirements must be “frozen” to early in the life cycle Requirements are validated too late

Applications(When to use?) Requirements are very well documented, clear and fixed. Product definition is stable. Technology is understood and is not dynamic. There are no ambiguous requirements. Ample resources with required expertise are available to support the product. The project is short.

* **Write phases of spiral model**

Ans - 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. learning with maturity and also involves minimum risk for the customer as well as the development firms. Following are the typical uses of Spiral model

When costs there are a budget constraint and risk evaluation is important. For medium to high-risk projects. Long-term project commitment because of potential changes to economic priorities as the requirements change with time. Customer is not sure of their requirements which are usually the case.

Requirements are complex and need evaluation to get clarity. New product line which should be released in phases to get enough customer feedback. Significant changes are expected in the product during the development cycle.