1. **What is SDLC?**

* Full form of SDLC is software development life cycle.
* The process for planning, implementation, testing, documentation, deployment and ongoing maintenance and support.
* SDLC phases :-
* Requirements collection/ gathering :- Establish customer needs
* Analysis :- Model and specify the requirements (what we need are)
* Desing :- Model and specify a solution
* Implementation :- Construct a solution in software
* Testing :- Validation the solution against the requirements
* Deployment :- To live software for users
* Maintenance :- Repair defect and adapt the solution to the new requirements.

1. **What is software testing?**

* Software testing is a process used to identify the correctness, completeness and quality of developed computer software.
* It can also be stated as the verification and validation.
* Testing activities :-
* Planning and control
* Choosing test
* Condition designing
* Test cases checking results
* Evaluating completion results
* Reporting on the testing process and system under test finalizing or closure
* Testing also includes reviewing of documents and static analysis.

Test objectives :-

* Finding defects
* Preventing defects
* Gaining confidence in and providing information about the level of quality.

1. **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 method break the product into small incremental builds.
* These builds are provided in iterations.
* Each iteration typically lasts from about one to three weeks or two to four weeks.
* Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.

1. **What is SRS?**

* SRS full form is software requirement specification.
* A software requirements specification is a complete description of the behaviour or 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 requirements.
* Non-functional requirements are requirements which impose constraints on the design or implementation.

1. **What is oops?**

* Full form is 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.

1. **Write Basic Concepts of oops.**

* Class :- Class is a structure in which we can have member functionality and member variables are there.
* Object :- Object is instance of class with state and behaviour.
* Inheritance :- To access property of one class to another class.
* Polymorphism :- Same function name but having different functionality.
* Abstraction :- Hiding internal detail and showing essential information to users.
* Encapsulation :- To wrapping data into single unit.

1. **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.
* That is both data and function that operate on data are bundled as a unit called as object.

1. **What is class?**

* Define a class is you define a blueprint for an object.
* A class represents an abstraction of the object and abstracts the properties and behaviour of that object.

1. **What is encapsulation?**

* Encapsulation enables data hiding, hiding irrelevant information from the users of a class and exposing only the relevant details required by the user.
* To wrapping data into single unit.

1. **What is inheritance?**

* Inheritance means that one class inherits the characteristics of another class.
* To access property of one class to another class.

1. **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.
* Same function name but having different functionality.

1. **Draw Usecase on online bill payment system (paytm).**

* [**https://app.diagrams.net/?splash=0#G1ztzhld9JjOIVkGDIo06ygDXxkH3dsIK5#%7B%22pageId%22%3A%22CO1hSf9II64KMC1ewBNL%22%7D**](https://app.diagrams.net/?splash=0#G1ztzhld9JjOIVkGDIo06ygDXxkH3dsIK5#%7B%22pageId%22%3A%22CO1hSf9II64KMC1ewBNL%22%7D)

1. **Draw Usecase on banking system for customers.**

* [https://app.diagrams.net/#G1U3lmAn3nYYuKwIsfyoGDF5HtT3pZ-D\_W#%7B%22pageId%22%3A%22\_64suvJ2hkUH1BiCplBC%22%7D](https://app.diagrams.net/#G1U3lmAn3nYYuKwIsfyoGDF5HtT3pZ-D_W)

1. **Draw Usecase on Broadcasting System.**

* [**https://app.diagrams.net/#G1LGJP7OGxB4rQ1D8X\_vBpWSrCL4hWlk7F#%7B%22pageId%22%3A%22E8qeh5xjLGWfdh4i7Xfc%22%7D**](https://app.diagrams.net/#%7B%22pageId%22%3A%22E8qeh5xjLGWfdh4i7Xfc%22%7D)

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

* Requirement gathering :-
* Features
* Usage scenarios
* Although requirements may be documented in written form, they may be incomplete, unambiguous, or even incorrect.
* Requirements will Change.
* There are two types of requirements :-

Functional and nonfunctional.

* Analysis :-
* The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished.
* This phase defines the problem that the customer is trying to solve.
* The deliverable result at the end of this phase is a requirement document.
* Desing :-
* Design Architecture Document Implementation Plan.
* Critical Priority Analysis Performance Analysis Test Plan.
* The requirement document must guide this decision process.
* Implementation :-
* In the implementation phase, the team builds the components either from scratch or by composition.
* Given the architecture document from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility.
* The implementation phase deals with issues of quality, performance, baselines, libraries, and debugging.
* Testing :-
* Simply stated, quality is very important.
* Many companies have not learned that quality is important and deliver more claimed functionality but at a lower quality level.
* A customer satisfied with the quality of a product will remain loyal and wait for new functionality in the next version.
* Quality is a distinguishing attribute of a system indicating the degree of excellence.
* Responsible for establishing high quality across all phases.
* Maintenance :-
* Software maintenance is one of the activities in software engineering, and is the process of enhancing and optimizing deployed software, as well as fixing defects.
* Software maintenance is also one of the phases in the System Development Life Cycle, as it applies to software development.
* The maintenance phase is the phase which comes after deployment of the software into the field.
* Corrective maintenance :- Identifying and repairing defects.
* Adaptive maintenance :- Adapting the existing solution to the new platforms.
* Perfective Maintenance: Implementing the new requirements.

1. **Explain Phases of the waterfall model.**

* Requirement :- The goal of this phase is to collect all the system requirement from the clients or users.
* Analysis :-
* Desing :- In this phase the system architecture and design are planned based on the requirements gathering in the previous phase.
* Implementation :- This is where the actual coding takes place based on the design specification.
* Testing :- After coding is complete the system tested to ensure it meets the defined requirements.
* Maintenance :- In this phase, any issues that arise are addressed. This phase involves updating the system, fixing bugs and making necessary adjustments over time.

1. **Write phases of spiral model.**

* Planning :- Determine the objectives, alternatives and constrints.
* Risk analysis :- Analysis of alterations and identifications/ resolution of risk.
* Engineering :- Development of next level product.
* Cuntomer evaluation :- Assessment of the result of engineering.

1. **Write agile manifesto principles.**

* Customer satisfaction
* Accept changing requirements
* Deliver working software frequently

1. **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.
* Pros :-
* Is a very realistic approach to software development Promotes team work and cross training.
* Functionality can be developed rapidly and demonstrated.
* Resource requirements are minimum.
* 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.
* There is very high individual dependency, since there is minimum documentation generated.

1. **Draw usecase on OTT Platform.**

* <https://app.diagrams.net/?splash=0#G1aM4G48uaChUa2AinMsiFTJihA2wqo8tt#%7B%22pageId%22%3A%22V-kM1FVtaYr_OpooYo49%22%7D>

1. **Draw usecase on E-commerce application.**

* <https://app.diagrams.net/#G1GRkmMrYw4WM7JcRyFeFs5HoRI8DY7F-Z#%7B%22pageId%22%3A%22p_-u5Uka6BaevqLyTCMw%22%7D>

1. **Draw usecase on Online shopping product using payment gateway.**

* [**https://app.diagrams.net/#G1s0spOuACzUQXdHVj2FAohH0TpPMzQPwe#%7B%22pageId%22%3A%22fK3qVjxC7wu0AX9WiV3x%22%7D**](https://app.diagrams.net/#G1s0spOuACzUQXdHVj2FAohH0TpPMzQPwe#%7B%22pageId%22%3A%22fK3qVjxC7wu0AX9WiV3x%22%7D)