**Q1. What is SDLC?**

A1. Software Development Life Cycle is cost effective and time management cycle that is used by development team to design and built high quality software. SDLC consists of 5 phases i.e. Planning, Analysis, Design, Implementation, Testing & Integration.

**Q2. What is Software Testing?**

A2. Software testing is the process that evaluates that the website or application is developed as per requirements, and it is defect free. The main benefit of software testing is it removes bugs and improves performance.

**Q3. What is agile methodology?**

A3. Agile methodology is combination of iterative and incremental model. It is way to manage a project by breaking it up into several phases. Agile methodology emphasizes on adapting on needs of the customer, team, and changes in your project requirements.

**Q4. What is SRS?**

A4. Software Requirement Specification is the document that describes what the software will do and how it is expected to perform. SRS gives you a complete picture of your entire project. It is your plan of action which keeps developers and testers on same page.

**Q5. What is Oops?**

A5. Object Oriented Programming is a programming model that organizes software design around data or object rather than function or logic. OOP focuses on the object that developers want to manipulate rather than logic required to manipulate them. OOPs approach is programming is well suited for large, complex and actively maintained or updated programs.

**Q6. Write basic concepts of Oops.**

A6.

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

**Q7. What is Object?**

A7. An object is anything to which a concept applies. For example: An object can be a car, an employee, a flight, a colour etc. There are two parts of objects Data + Methods.

**Q8. What is class?**

A8. Class is defined as a blueprint of an object. A class is a group of objects that share common properties and behaviour.

**Q9. What is Encapsulation?**

A9. 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. In simple words it means “to enclose” like it is in capsule. Encapsulation enables data hiding, hiding irrelevant information from users of a class and exposing only the relevant details required by the users.

**Q10. What is inheritance?**

A10. Inheritance means that one class inherits the characteristics of another class. This is also called a “is a” relationship. 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. For example, Grand Parent > Parent > Child

**Q11. What is Polymorphism?**

A11. Polymorphism means “having many forms”. There is two types of polymorphism in Java

- Compile time polymorphism (Overloading)
- Runtime polymorphism (Overriding)

Q12.

**Q14. Write SDLC phase with basic introduction?**

A14. There are seven phases of SDLC.

- **Planning** - This is the phase where developer will plan for the upcoming project.
- **Analysis** – In Analysis stage concept is to gather all the specific details require for the new project as well as to determine first idea of prototype. At this stage developers will often create SRS document.
- **Design**- At this stage developer will outline all the details of overall application. Such as user interface, system interface, network requirements and Database
- **Development**- The development stage is the part where developers write code and build the application according to the earlier design documents and outlined specifications.
- **Testing** – Building software is not just the end. Now its time to check errors and bugs and make sure that end user application is not negatively affected at any point. It is important to check that software overall ends up meeting quality standards that were defined in SRS document.
- **Implementation** – After testing the overall design of the software will come together. Different modules or design will be integrated into primary source code by developers. After passing this stage software is theoretically ready for market and can be used by any end user.
- **Maintenance** - In this stage developers are responsible for implementing any changes that the software might need after deployment.

**Q15. Explain phases of waterfall model.**

A15. Waterfall model is the sequential development process that flows like a waterfall through all phases on project (Analysis, design, development and testing) with each phase completely wrapping up before moving to next phase.

**Q16. Write phases of Spiral model.**

A16. The Spiral methodology is known for being one of the most flexible SDLC models. It has a repetitive cycle—the project goes through four stages (planning, risk analysis, engineering, and evaluation) over and over in a “spiral” formation until it is fully completed. The spiral model is a systems development lifecycle (SDLC) method used for risk management that combines the iterative development process model with elements of the Waterfall model.

**Q17. Write agile manifesto principle.**

A17.

- **Individuals and interactions** - in agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
- **Working software** - Demo working software is considered the best means of communication with the customer to understand their requirement, instead of just depending on documentation.
- **Customer collaboration** - As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
- **Responding to change** - agile development is focused on quick responses to change and continuous development.

**Q18. Explain working methodology of agile model and write pros and cons.**

A18. 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.

| Pros | Cons |
|---|---|
| 1. Is a very realistic approach to software development.<br>2. Promotes teamwork and cross training.<br>3. Functionality can be developed rapidly and demonstrated.<br>4. Resource requirements are minimum.<br>5. Suitable for fixed or changing requirements.<br>6. Delivers early partial working solutions.<br>7. Good model for environments that change steadily.<br>8. Minimal rules, documentation easily employed.<br>9. Enables concurrent development and delivery within an overall planned context.<br>10. Little or no planning required.<br>11. Easy to manage.<br>12. Gives flexibility to developers | 1. Not suitable for handling complex dependencies.<br>2. More risk of sustainability, maintainability, and extensibility.<br>3. An overall plan, an agile leader and agile PM practice is a must without which it will not work.<br>4. Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.<br>5. Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.<br>6. There is very high individual dependency, since there is minimum documentation generated.<br>7. Transfer of technology to new team members may be quite challenging due to lack of documentation |