1. *What is SDLC*

Ans.> SDLC is software development life cycle ,SDLC is a structure imposed on the development of

a software product defines the process for palninig impelmentation testing documentation deployment and on going

maintenance and support .

2. What is software testing?

Ans.> software testing is process used to identify the correctness ,completness and quality of developed computer software

4. What is SRS

Ans.> SRS is sofware requirement specification details the specific requirements of the software that is to be develpoed

5. What is oops

Ans.> oop is Object Oriented Programming

6. What is oops

Ans.> oop means object oriented programming

oop 6 types

1.object

2.class

3.abstraction

4.encapsulation

5.inhertanc

6.polmorphism

7. What is object

Ans.> object: any entity which has own state and behaviour

ex pen paper

8. What is class

Ans.> class:collection of object

ex human body

9. What is encapsulation

Ans.> encapsulation: wrapping of data or binding of data

ex apsule

10. What is inheritance

Ans.> inhertance: when one object acquire all the properties and behaviour of parent classe

ex parent child

11. What is polymorphism

Ans.> polymorphism: many way to performa anything

ex overloding ,overiding

12. Write SDLC phases with basic introduction

Ans.> SDLC IS is a structure imposed on the development of a software product defines the process for palninig impelmentation testing documentation deployment and on going maintenance and support .

SDLC phases

1 Requirements Gathering:Establish Customer Needs

2 Analysis:Model And Specify the requirements-what

3 Design:Model And Specify a Solution – Why

4 Implementation:Construct a Solution In Software

5 Testing:Validate the solution against the requirements

6 Maintenance:Repair defects and adapt the solution to the new requirements

13. Explain Phases of the waterfall model

Ans.> waterfall model is the classical software lifecycle models the software development as a step-by step

waterfall betwwen the various development phases

requirement must be frozen to early in the life cycle

requirement are validated too late

1.requirement

2.analusis

3.design

4.implementation

5.testing

6.maintenance

14. 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.

17. Write agile manifesto principles

*Ans.> * 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.*

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

**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.*