1. What is SDLC

**SDLC** - **Software Development Life Cycle**. It is a step-by-step process used to develop software. SDLC helps to make sure that the software is well-planned, properly designed, tested, and delivered on time.

SDLC phases

1. **Requirement Gathering** – understanding what the customer needs
2. **Planning** – preparing time, cost, and resources
3. **Design** – designing how the software will work
4. **Development** – writing the code
5. **Testing** – checking if the software works properly
6. **Deployment** – giving the software to the customer
7. **Maintenance** – fixing problems after the software is released
8. What is software testing?

Software Testing means checking the software to see if it is working correctly or not.

It helps to find bugs in the software before it goes to the user.

The main aim is to make sure:

* The software gives the correct result
* It is easy to use
* It does not crash or stop working
* It is safe to use

There are two types of testing:

1. Manual Testing – Testing by hand, without any tools
2. Automation Testing – Testing using software tools
3. What is Agile Methodology ?

Agile Methodology is a way to make software **step by step** in **small parts**. Each part is made, tested, and shown to the customer quickly.

1. It helps to:
2. Work fast
3. Make changes easily
4. Give better results
5. Scrum Meeting
6. What is SRS

SRS - Software Requirements Specification.  
It is a document that explains what the software should do and how it should work.

The SRS is prepared before the software development starts. It includes:

* Features the software must have
* How the software should behave
* Rules the software must follow
* What users need from the software

The SRS is helpful for:

* Developers (to build the software)
* Testers (to test if it works correctly)
* Clients (to check if their needs are covered)

1. What is oops

OOPs - Object-Oriented Programming System.

It is a way of writing programs using objects. Objects are like small parts that have data and functions together.

* Class: A template to create objects
* Object: A real thing created from a class
* Inheritance: Reusing code from another class
* Encapsulation: Hiding data inside an object
* Polymorphism: One thing behaving in many ways

1. Write Basic Concepts of oops

There are 4 main concepts of Object-Oriented Programming (OOPs):

1.Class

2.Object

3.Inheritance

4.Encapsulation

5.Polymorphism

1. What is object

 object - data store - multiple data store   
       object is one kind of variable which store multiple data // class   
   variable - data store at a time single store

1. What is class

* collection of method, object, datatypes and constructor and variables

1. What is encapsulation

Encapsulation means hiding the internal details of an object and only showing what is necessary.

It keeps the data safe from outside changes by using private variables and public methods to access them.

1. What is inheritance

Inheritance means one class can take or reuse the features of another class.

It helps to:

* Avoid writing the same code again
* Reuse existing code
* Make programs easier to manage

1. What is polymorphism

**Polymorphism** means **one thing behaving in different ways**.In programming, it allows **the same function** to do **different tasks** based on input or class.

Polymorphism means using one name for many actions.  
It helps make the code flexible and reusable.

1. Write SDLC phases with basic introduction

**Phases of SDLC:**

1. **Requirement Gathering & Analysis** – Understanding what the client or market needs.
2. **Planning** – Defining scope, resources, timelines, and cost estimation.
3. **Design** – Creating architecture and design specifications (e.g., UI design, system architecture).
4. **Development** – Writing the actual code based on design documents.
5. **Testing** – Ensuring the software is bug-free and meets requirements through QA testing.
6. **Deployment** – Releasing the software for use in the production environment.
7. **Maintenance** – Ongoing support, updates, and bug fixing after deployment.
8. Explain Phases of the waterfall model

The Waterfall Model is the oldest and simplest model in SDLC.  
In this model, work is done step by step, like water flowing down — one phase is completed before the next starts.

Phases of the Waterfall Model:

1. Requirement Gathering:  
   Understand what the client wants from the software. All requirements are written clearly.
2. System Design:  
   Plan how the software will look and work. This includes screen design, database, and system layout.
3. Implementation (Coding):  
   Developers write the actual code based on the design.
4. Testing:  
   The testing team checks the software for bugs and errors.
5. Deployment:  
   The software is given to the client or made live for users.
6. Maintenance:  
   If there are any problems or updates needed after delivery, they are fixed in this phase.
7. What is Exploratory Testing?

Exploratory Testing means checking the software freely without following written steps

In this type of testing, the tester:

* Learns the application
* Explores how it works
* Tests it freely using their own ideas and experience

This is useful when:

* There is less time
* No documents are available
* The tester wants to find hidden bugs

1. What is Integration testing?

Integration Testing is the type of testing where two or more parts (modules) of software are combined and tested together.

The goal is to check:

* If the modules work well together
* If data is passed correctly between them
* If any error happens when modules are connected

1. What is Alpha testing?

Alpha Testing is the type of testing done by the testing team inside the company before the software is released to users.

It is done to:

* Find bugs
* Check if the software works as expected
* Make sure it is ready for real users

1. What is Beta testing?

Beta Testing is done by real users outside the company before the final release of the software.

The software is given to selected users to:

* Use it in real conditions
* Find any issues or bugs
* Give feedback to the company

1. What isfunctional system testing?

Functional System Testing means **checking the full software to see if all features work properly.**

* All features (like login, search, payment)
* If the system gives the correct output for given input
* If the software behaves as expected

Only the functionality is tested — not the internal code.

1. What is Non-Functional Testing?

Non-Functional Testing means testing the quality of the software — like speed, safety, and user-friendliness — not the actual features

 Speed (how fast the software runs)

 Performance (how it works under load)

 Security (how safe it is)

 Usability (how easy it is to use)

1. What is GUI Testing?

GUI Testing means testing the Graphical User Interface of the software.  
It checks how the software looks and feels to the user.

It includes checking:

* Buttons, menus, icons
* Text boxes, colors, and font size
* Layout and screen design
* Mouse clicks and keyboard actions

1. What is load testing?

Load Testing means testing the software by giving it **heavy load** to see if it still works properly.

 How the software works when many users use it together

 If the system slows down or crashes under pressure

 The maximum limit of the software’s performance

1. What is stress Testing?

Stress Testing is a type of testing where the software is tested beyond its normal limits to see how it behaves under extreme conditions.

It helps to check:

* What happens when the system is overloaded
* If the software can handle failure
* Whether it recovers after crashing

1. What is white box testing?

White Box Testing is a type of software testing where the internal code of the application is tested.

In this testing, the tester needs to know the programming and logic of the code.

It is used to:

* Check if all paths in the code are working
* Test how the program makes decisions
* Find hidden errors in the code

1. What is black box testing?

Black Box Testing is a type of testing where the tester checks the functionality of the software without looking at the internal code.

The tester only checks:

* If the software gives the correct output
* If all features like login, search, payment are working properly
* How the system behaves in different situations

1. What is 7 key principles? Explain in detail?

1.testing shows the presence of defects

- Testing can reveal defects, but it cannot prove a software is completely defect-free.

2.Exhaustive testing is impossible

Due to the vast number of possible input combinations and conditions, it’s not feasible to test every scenario.

3. Early testing

Starting testing early in the development lifecycle helps identify and fix defects sooner, reducing costs and effort.

4.Defect clustering

A small subset of modules or functionalities often contains the majority of defects.

5. pesticide paradox

Repeating the same tests can become less effective at finding new defects over time, requiring test cases to be regularly reviewed and updated.

6. Testing is context dependent

The testing approach should be tailored to the specific project, system, and context.

7. Absence of errors fallacy

- If a built software is 99% bug-free it does not follow user requirement then it is unusable. It is not only necessary that software is 99% bug-free but it is also mandatory to fulfil all the customers requirements.

1. Difference between QA v/s QC v/s Tester

Quality Assurance (QA) is the process of making sure that the software development process is done correctly and with quality.

QA focuses on:

* Preventing defects
* Following rules and standards
* Making sure the right methods are used to build the software

QA does not test the software directly, but it helps the team to build quality software from the beginning.

Quality Control (QC) is the process of checking the final software to find any defects or bugs before it is released to users.

QC focuses on:

* Finding and fixing defects in the product
* Making sure the software meets the customer’s needs
* Testing the product after development

QC is done after the software is developed, usually by test

1. Difference between Smoke and Sanity?

Smoke Testing is a type of software testing where the main features of the application are tested to check if the build is stable or not.

It is done after a new build is released to make sure:

* The software opens properly
* Basic functions like login, homepage, search are working
* There are no major issues

If smoke testing fails, the build is rejected for further testing.

Sanity Testing is a type of software testing done to check if a small change or bug fix in the software is working correctly.

It is done after minor changes to:

* Make sure the fixed feature works fine
* Check if the change has not affected other parts
* Decide whether to continue with further testing

Sanity testing is narrow and deep – it checks a small area in detail.

1. Difference between Priority and Severity

**Priority** means how **quickly a bug should be fixed** based on **business or user needs**.It shows the **importance of fixing the bug** from the customer's point of view.

Severity means how serious or harmful a bug is to the functionality of the software.It shows the impact of the defect on the working of the system.

1. What is Bug Life Cycle?

Bug Life Cycle is the process a bug goes through, from the time it is found to the time it is fixed and closed.

Each bug moves through different statuses in its life.

🔄 Stages of Bug Life Cycle:

1. New – The tester finds a new bug
2. Assigned – The bug is given to a developer to fix
3. Open – The developer starts working on the bug
4. Fixed – The developer fixes the bug
5. Retest – The tester checks if the bug is really fixed
6. Closed – The bug is fixed and working fine
7. Reopened – If the bug is still there after testing
8. Rejected / Not a Bug – If the bug is invalid or not a real issue
9. Deferred – The bug is postponed to fix later
10. Explain the difference between Functional testing and NonFunctional testing

Functional Testing is done to check whether the features of the software are working as expected.

It checks:

* What the software does
* If the software gives the correct output
* If all buttons, forms, and functions are working

Functional Testing is done to check whether the features of the software are working as expected.

It checks:

* What the software does
* If the software gives the correct output
* If all buttons, forms, and functions are working

1. What is RDBMS

Relational database management system

* It is database system that store data in table and row
* Mysql, oracle,postgrey, sqlite

32.What is SQL

SQL stands for Structured Query Language.  
It is used to store, manage, and retrieve data from a database.

Testers use SQL to:

* Check data in the database
* Write queries to find or update information
* Validate test results with actual data

33.Write SQL Commands

1. SELECT

Used to get data from a table.

SELECT \* FROM employees;

2. INSERT

Used to add new data into a table.

INSERT INTO employees (name, age, city)

VALUES ('Amit', 28, 'Delhi');

3. UPDATE

Used to change existing data in a table.

UPDATE employees

SET city = 'Mumbai'

WHERE name = 'Amit';

4. DELETE

Used to remove data from a table.

DELETE FROM employees

WHERE name = 'Amit';

5. CREATE TABLE

Used to create a new table.

CREATE TABLE employees (

id INT,

name VARCHAR(50),

age INT,

city VARCHAR(50)

);

6. DROP TABLE

Used to delete a table completely.

DROP TABLE employees;

7. WHERE

Used to add conditions in queries.

SELECT \* FROM employees

WHERE city = 'Mumbai';

34. What is join?

A JOIN in SQL is used to combine data from two or more tables based on a related column.

It helps to get complete information from different tables.

35. Difference between RDBMS vs DBMS

1. DBMS (Database Management System)  
DBMS is software that helps to store, manage, and retrieve data from a database.

It can store data in files, but does not support relationships between tables.

2. RDBMS (Relational Database Management System  
RDBMS is an advanced version of DBMS that stores data in tables and allows relationships between those tables using keys (like primary key, foreign key).

It follows rules of relational models and is used in most modern systems.

* MySQL
* Oracle
* SQL Server

36. Write a query to create the table in Structured Query Language.

CREATE TABLE Employees (

EmpID INT PRIMARY KEY,

Name VARCHAR(50),

Age INT,

City VARCHAR(50),

Salary INT

);

37. Write a query to insert data into table.

INSERT INTO Employees (EmpID, Name, Age, City, Salary)

VALUES (1, 'Rahul', 30, 'Delhi', 50000);

38. Write a query to update data into table with validations.

UPDATE Employees

SET Salary = 60000

WHERE Age > 25 AND EmpID = 1;

39. Write a query to delete data from table with validations.

DELETE FROM Employees

WHERE City = 'Delhi' AND EmpID = 1;

40. Query to Drop Table and Database

Drop table

DROP TABLE Employees;

Drop database

DROP DATABASE CompanyDB;

40. What is API Testing

API Testing means testing the Application Programming Interface (API) to check if it works correctly.

APIs are used to let two software systems talk to each other. In API Testing, we test if the data is sent and received correctly between them.

If the API response is correct

* If the data format (JSON, XML) is right
* The speed and performance of the API
* If the API works for valid and invalid inputs

41. Types of API Testing

1. Functional Testing

Checks if the API is working correctly as per the requirements.  
Example: Does the login API return the correct user info?

2. Performance Testing

Checks how fast and stable the API is under different conditions.  
Example: Does the API respond quickly when 100 users access it?

3. Load Testing

Checks how the API works when many users use it at the same time.  
 Example: 1000 people use the payment API at once.

4. Security Testing

Checks if the API is safe from hackers and only authorized users can access it.  
Example: Can someone access user data without login?

5. Validation Testing

Checks if the response data is in the correct format and values are correct.  
 Example: Is the date in correct format like YYYY-MM-DD?

6. Error Testing (Negative Testing)

Checks how the API behaves when given wrong input or missing values.  
Example: What happens if we send an empty password?

7. Integration Testing

Tests how the API connects with other systems or modules.

42. What is Responsive Testing?

Responsive Testing is a type of testing used to check if a website or application looks and works properly on different screen sizes and devices.

It ensures that the content adjusts correctly on:

* Mobile phones
* Tablets
* Laptops
* Desktop
* Layout and design on different screens
* Font size and button alignment
* Scroll, zoom, and navigation behavior
* Images and content visibility

43. What is Automation Testing?

Automation Testing is a type of software testing where test cases are executed using tools or scripts instead of doing it manually.

* Save time by running tests faster
* Repeat tests automatically
* Test large applications quickly
* Reduce humanerrors

44. What are the benefits of Automation Testing?

1. Saves Time

Tests run faster than manual testing, especially for big projects.

2. Reusable Test Scripts

Once a script is written, it can be used again and again for the same tests.

3. Increases Test Coverage

You can test more features and cases in less time.

4. Reduces Human Errors

Automated tests follow the same steps every time, so there’s no chance of mistake.

5. Supports Regression Testing

Very useful when testing the same features repeatedly after every update.

6. Saves Cost in Long Term

Though setup is expensive at first, it saves money over time by reducing manual work.

7. Better Accuracy and Reporting Automated tools give detailed test reports and results instantly.