AASSIGNMENT

1. WHAT IS SDLC?

Sdlc Software Development Life Cycle is a process used by software industry to design develop and test high quality software

The SDLC aims to produce a high quality software that meets customer expectations

Sdlc is six phase.

1.Requirement gathering analysis

Understand what the customer needs.

Gather all funcation non funcation requirements

Create regular specification document

2 Degine

Create system architecture and degine documents.

Decide technology database and ui degine

Two level degine

1.high level degine

2.low level degine

3 Implementation/coding

Developers write the actual code based on the degine.

4 Testing

Tester the softwar testing for bugs and defects.

Four type testing'

Unit testing

Integration testing

System testing

Uat testing

5 Deployment

Release the softwar live environment sometimes done in phase

6 Maintence

Customer problem solve and website uadate ,editing solve the problem

AASSIGNMENT

2. WHAT IS SOFTWARE TESTING?

Software Testing is a part of software development process

Software Testing is an activity to detect and identify the defects in the software

The objective of testing is to release quality product to the client

3. WHAT IS OPPS?

The full form of Oops is Object-Oriented Programming System
It is a programming model or paradigm that organizes software design around objects, rather than functions and logic.

In OOPS objects are

Instance of class, contain data , contain functions that operate on the data

4. WRITE BASIC CONCEPTS OF OOPS?

Main Concepts of OOPS

Class –   (A blueprint for creating objects.) It defines attributes (variables) and behaviors (methods)

Object – A real-world entity created from a class.

Encapsulation – Bundling data and methods that operate on the data within one unit (class).

Inheritance – A way to create new classes from existing ones, allowing code reuse.

Polymorphism – The ability to use a common interface for multiple forms

Example   method overloading/overriding

Abstraction – Hiding complex details and showing only the essential features of the object.

1. WHAT IS OBJECT?

An object is an instance of a class that contains data (attributes) and behaviors (methods). It represents a real-world entity

An object is an instance of a class.

It represents a real-world entity with state and behavior.

2. WHAT IS CLASS?

AASSIGNMENT

A class is a user-defined data type that acts as a blueprint for creating objects
It defines the attributes (also called fields or properties) and methods (functions) that the created objects will have

A class is a blueprint or template for creating objects
It defines variables And behaviors methods

3. WHAT IS ENCAPSULATION ?

Encapsulation is the object-oriented programming concept of binding data (variables) and the methods (functions) that operate on that data into a single unit, called a class**.**

Wrapping data (variables) and methods (functions) into a single unit (class).
Protects data from outside access by using access modifiers (like private or public).

4. WHAT IS INHERITANCE ?

Inheritance is an OOP concept where a child class (subclass) inherits properties and methods from a parent class (superclass), enabling code reuse and hierarchical classification**.**

One class (child) can inherit the properties and methods of another class (parent)

5. WHAT IS POLYMORPHISM ?

Polymorphism is an OOP concept that allows one interface to be used for different forms or behaviors**,** meaning the same method name can perform different tasks depending on the context.

6. WRITE SDLC PHASES WITH BASIC INTRODUCTION ?

SDLC (Software Development Life Cycle) is a structured process used for developing software applications. It outlines a series of phases that guide the development process to ensure the software is built systematically and efficiently.

7. WHAT IS FUNCTIONAL SYSTEM TESTING ?

Functional System Testing is a type of software testing that focuses on verifying that the entire system behaves according to the specified functional requirements**.**

Object Properties Coverage
Input Domain Coverage (BVA, ECP)
Database Testing/Backend Coverage
Error Handling Coverage
Calculations/Manipulations Coverage
Links Existence & Links Execution
Cookies & Sessions

8. WHAT IS NON-FUNCTIONAL SYSTEM TESTING ?

AASSIGNMENT

Non-Functional System Testing is a type of software testing that focuses on evaluating the non-functional aspects of a software application — such as performance, usability, reliability, security, and scalability.

9. WHAT IS GUI TESTING ?

There are two types of interfaces in a computer application.
Command Line Interface is where you type text and computer responds to that command.
GUI stands for Graphical User Interface where you interact with the computer using images rather than text

10. WHAT IS LOAD TESTING ?

Testing speed of the system while increasing the load gradually till the customer expected number.
Load Testing is a type of non-functional software testing that checks how a system performs under expected user load.

11. WHAT IS INTEGRATION TESTING?

In Integration Testing, individual software modules are integrated logically and tested as a group.
Integration testing focuses on checking data communication amongst these modules. Integrated Testing is white box testing technique.
Integrated testing is conducted by the developers.
Approaches:
Top Down Approach
Bottom Up Approach
Sandwich Approach(Hybrid)

12. WHAT IS STRESS TESTING?

Testing speed of the system while increasing/reducing the load on the system to check any where its breaking.

Stress Testing is a type of non-functional testing that evaluates how a software system behaves under extreme or breaking-point conditions such as very high user traffic, limited memory, or CPU usage — to determine its stability and error handling under stress.

13. WHAT IS A WHITE BOX TESTING AND LIST THE TYPE OF WHITE BOX TESTING?

White Box Testing conducts on internal logic of the programs.
Programming Skills are required.

Ex: Unit Testing & Integration Testing

AASSIGNMENT

14. WHAT IS A BLACK BOX TESTING.WHAT ARE THE DIFFERENT BLACK BOX TESTING TECHNIQUES?

Testing conducts on functionality of the application whether it is working according to customer requirements or not.

Ex: System Testing & UAT Testing

   GUI TESTING
   USABILITY TESTING
   FUNCTIONAL TESTING
   NON-FUNCTIONAL TESTING

15. WHAT IS A 7 KEY PRINCIPLE ? EXPLAIN IN DETAILS ?

Seven key principle

   1   Testing shows the Presence of Defects

   The goal of software testing is to make the software fail. Software testing reduces the presence of defects. Software testing talks about the presence of defects and doesn't talk about the absence of defects. Software testing can ensure that defects are present but it can not prove that software is defect-free. Even multiple tests can never ensure that software is 100% bug-free. Testing can reduce the number of defects but not remove all defects.

   2   Exhaustive Testing is not Possible

    Test case login positive and negative
 correct email and correct pass - login successful home page
 correct email and incorrectpass incorrectpass
 incorrect email and password correct incorrect email
 incorecrt email and incorrect pass incorrect mail and pass Precondistion

It is the process of testing the functionality of the software in all possible inputs (valid or invalid) and pre-conditions is known as exhaustive testing. Exhaustive testing is impossible means the software can never test at every test case. It can test only some test cases and assume that the software is correct and it will produce the correct output in every test case. If the software will test every test case then it will take more cost, effort, etc., which is impractical.

3 Early Testing

To find the defect in the software, early test activity shall be started. The defect detected in the early phases of SDLC will be very less expensive. For better performance of software, software testing will start at the initial phase i.e. testing will perform at the requirement analysis phase.

4 Defect Clustering Cluster

In a project, a small number of modules can contain most of the defects. The Pareto Principle for software testing states that 80% of software defects come from 20% of modules.

5 Pesticide Paradox

Repeating the same test cases, again and again, will not find new bugs. So it is necessary to review the test cases and add or update test cases to find new bugs.

6 Testing is Context-Dependent

The testing approach depends on the context of the software developed. Different types of software need to perform different types of testing. For example, The testing of the e-commerce site is different from the testing of the Android application.

7 Absence of Errors Fallacy

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

16. DIFFERENCE BETWEEN SMOKE AND SANITY ?

SMOKE TESTING

This is also basic functional testing conducted by developer or tester before releasing the build to the next cycle.

SANITY TESTING

This is a basic functional testing conducted by test engineer whenever receive build from development team.

AASSIGNMENT

SMOKE TESTING	SAINITY TESTING
Verify basic build stability	Verify specific functionality or fix
Broad, covers major features	Narrow focuses on specific areas
Shallow testing	Deep testing of selected areas
After receiving a new build	After changes or bug fixes in a stable build
Often automated	Usually manual
Can lead to rejecting a build	Generally does not reject the build

17. WHAT IS ADHOC TESTING?

Software testing performed without proper planning and documentation.
Testing is carried out with the knowledge of the tester about the application and the tester tests randomly without following the specifications/requirements.

18. WHEN SHOULD REGRESSION TESTING BE PERFORMED?

It is a process of identifying various features in the modified build where there is a chance of getting side-effects and retesting these features.
The new functionalities added to the existing system (or) modifications made to the existing system .
It must be noted that a bug fixer might introduce side-effects and a regression testing is helpful to identify these side effects.

19. DIFFERENCE BETWEEN VERIFICATION AND VALIDATION?

Verification checks whether we are building the right system.
Verification typically involves.
    Reviews
    Walkthroughs
    Inspections
Validation checks whether we are building the system right
Takes place after verifications are completed
Validation typically involves actual testing
System Testing

20. EXPLAIN TYPES OF PERFORMANCE TESTING?

   Performance testing is a three types.

1.Load:
   Testing speed of the system while increasing the load gradually till the customer expected number.

AASSIGNMENT

2.Stress:

Testing speed of the system while increasing/reducing the load on the system to check any where its breaking.

3.Volume:

Check how much volumes of data is able to handle by the system