**SOFTWARE TESTING ASSIGNMENT**

**MODULE-1**

* **What is SDLC?**

SDLC stands for software development life cycle is a essentially

A series of steps, or phase, that provide model for the development

And life cycle management of an application piece of software.

In short SDLC is structure imposed on the development of software product that defines the process for planning, implementation, testing, documentation, and deployment and ongoing maintains and support.

SDLC phases

1. Requirements collection/ gathering

2. Analysis

3. Design

4. Implementation

5. Testing

6. Maintenance.

* **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.

1. Agile model believes that every project needs to be handled

Differently and existing methods.

In agile the tasks are divide to time boxes (small time frames)

To deliver specific features to be release.

2. Iterative approach is taken and working software build is delivered after each iteration .Each build is incremental in terms of features, the final build holds all the features required

By the customer.

3. Agile through process had started early in the software development and started becoming popular with the time due

To its flexibility and adaptability.

* **What is SRS?**

A Software Requirements Specification (SRS) is a

Complete description of the behavior 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 requirement.in addition to use cases, the SRS also contains Non-functional requirements.

**Types of Requirements:**

1. Customer Requirements
2. Functional Requirements
3. Non-functional Requirements

* **What is OPPS?**

**Object Oriented Programming** is a computer programming model that organizes software design around data, or objects, rather than functions and logic.

**An object is like a Black Box.**

* **Write Basic Concept of OOPs?**

The basic Object-oriented programming concepts are:

* **Object:-** An objects represents individual ,identifiable,item,unit, entity, either real

Or real or abstract, with a well –defined role in the problem domain.

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

* **Class** − A class is a data-type that has its own members i.e. data members and member functions. It is the blueprint for an object in object oriented programming language. It is the basic building block of object oriented programming in c++.
* **Inheritance:** Inheritance can be defined as the process where one (parent/super) class acquires the properties (methods and fields) of another (child/sub). With the use of inheritance, the information is made manageable in a hierarchical order.
* **Polymorphism**: Polymorphism is the ability of an object to perform different actions (or, exhibit different behaviours) based on the context.

1. Compile time polymorphism (overloading)

2. Runtime polymorphism (overriding)

* **Abstraction:** Abstraction is a process of hiding the implementation details from the user, only the functionality will be provided to the user. In other words, the user will have the information on what the object does instead of how it does it.
* **Encapsulation:** In object oriented programming, encapsulation is the concept of wrapping together of data and information in a single unit. A formal definition of encapsulation would be: encapsulation is binding together the data and related function that can manipulate the data.
* **Polymorphism** The name defines polymorphism is multiple forms. Which means polymorphism is the ability of object oriented programming to do some work using multiple forms. The behaviour of the method is dependent on the type or the situation in which the method is called.
* **What is object?**

 An object is an instance of a class. It is an entity with characteristics and behaviour that are used in the object oriented programming. An object is the entity that is created to allocate memory. A class when defined does not have memory chunk itself which will be allocated as soon as objects are created.

Syntax ==class name object name;

* **What is class?**

A class is a data-type that has its own members i.e. data members and member functions. It is the blueprint for an object in object oriented programming language. It is the basic building block of object oriented programming in c++. The members of a class are accessed in programming language by creating an instance of the class.

Some important properties of class are −

* **Class** is a user-defined data-type.
* A class contains members like data members and member functions.
* **Data members** are variables of the class.
* **Member functions** are the methods that are used to manipulate data members.
* Data members define the properties of the class whereas the member functions define the behaviour of the class.

A class can have multiple objects which have properties and behaviour that in common for all of them.

## **Syntax**

class class\_name {

   data\_type data\_name;

   return\_type method\_name(parameters);

}

* **What is Encapsulation?**

By definition, encapsulation describes the idea of bundling data and methods that work on that data within one unit, like a class in Java. This concept is also often used to hide the internal representation, or state of an object from the outside. This is called [information hiding](https://en.wikipedia.org/wiki/Encapsulation_(computer_programming)" \l "An_information-hiding_mechanism" \t "_blank).

Due to the concept of encapsulation in object oriented programming another very important concept is possible, it is data abstraction or Data Hiding. it is possible as encapsulating hides the data at show only the information that is required to be displayed.

* **What is Inheritance?**

Inheritance is a mechanism in which one class acquires the property of another class. For example, a child inherits the traits of his/her parents. With inheritance, we can reuse the fields and methods of the existing class. Hence, inheritance facilitates Reusability and is an important concept of OOPs.

1. Single inheritance

2. Multiple inheritance

3. Multi-level inheritance

4. Hierarchical inheritance

5. Hybrid inheritance

* **What is polymorphism?**

The name defines polymorphism is multiple forms. which means polymorphism is the ability of object oriented programming to do some work using multiple forms. The behaviour of the method is dependent on the type or the situation in which the method is called.

Let’s take a real life example, A person can have more than one behaviour depending upon the situation. like a woman a mother, manager and a daughterAnd this define her behaviour. This is from where the concept of polymorphism came from.

In c++ programming language, polymorphism is achieved using two ways. They are operator overloading and function overloading.

**Operator overloading** In operator overloading and operator can have multiple behaviour in different instances of usage.

**Function overloading** Functions with the same name that can do multiple types based on some condition.

* **What is RDBMS?**

A relational [database](https://www.techtarget.com/searchdatamanagement/definition/database) management system (RDBMS) is a collection of programs and capabilities that enable IT teams and others to create, update, administer and otherwise interact with a [relational database](https://www.techtarget.com/searchdatamanagement/definition/relational-database). RDBMS store data in the form of tables, with most commercial relational database management systems using [Structured Query Language](https://searchsqlserver.techtarget.com/definition/SQL) (SQL) to access the database.

* **What is SQL?**

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgrad and SQL Server use SQL as their standard database language.

Also, they are using different dialects, such as −

* MS SQL Server using T-SQL,
* Oracle using PL/SQL,
* MS Access version of SQL is called JET SQL (native format) etc.
* **Write SQL commands**

**1.Dala-dala Definition Language**

**Create -create a new table,or other object in database**

**Alter-modifies an existing database object,such as table.**

**Drop-deletes an entire table.**

**2.DML- Data Manipulation Language**

**Insert- creates a record**

**Update-modifies record**

**Delete-deletes records**

**3.DCL- Data Control Language**

**Grant-gives a privilege to user**

**Revoke -takes back privilege granted from user.**

**4. DQL- Data Query Language**

**Select-retrieves certain records from one or more tables.**

* **Draw Use case on online book shopping?**

Open browser

Search for book option

Select book

Add to cart

Place order

Sign in/registration

Done payment

enter your address

* **Draw use case on online bill payment System(paytm)**

Open google pay option

Login /rgistration

Bill pay option

Select electricity bill

Select city

Link account

Pay bill

Exit

* **Write SDLC phases with basic introduction**

SDLC is a structure imposed on the development

Of a software product that defines the process

For planing,testing,documentation,and ongoing maintain.

SDLC Phases:

1. Requirements collection/Gathering

Establish customer needs

1. Analysis

Model And Specify the requirement-”what”

1. Design

Model And specify in solution-”why”

1. Implementation

Construct a solution in software

1. Testing

Validate the solution against the requirements.

1. Maintenance

**Repair defects and adapt the solution to the new requirements.**

* **Explain Phases of the waterfall model**

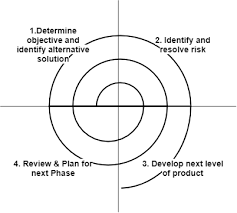
The waterfall model has six stages: **requirements, analysis, design, coding, testing, and deployment**. During the requirements stage, developers write down all the possible requirements of a system in a requirements document.



* **Write phases of Spiral Model**

The spiral model has four phases: **Planning, Design, Construct and Evaluation**. A software project repeatedly passes through these phases in iterations (called Spirals in this model).

The spiral model enables gradual releases and refinement of a product through each phase of the spiral as well as the ability to build prototypes at each phase. The most important feature of the model is **its ability to manage unknown risks after the project has commenced**; creating a prototype makes this feasible.



* **Write agil manifesto principles.**

# **The Agile Manifesto**

We are uncovering better ways of developing software by doing it and helping others do it.  
Through this work we have come to value:

****Individuals and interactions****over processes and tools

****Working software**** over comprehensive documentation

****Customer collaboration**** over contract negotiation

****Responding to change**** over following a plan

That is, while there is value in the items on the right, ****we value the items on the left more.****

* ****What is join ?****

****A join clause is used to combine row from two or more table ,based on related column between them.****

* ****Write type of join ?****

**SQL Join** statement is used to combine data or rows from two or more tables based on a common field between them. Different types of Joins are as follows:

* INNER JOIN
* LEFT JOIN
* RIGHT JOIN
* FULL JOIN

### ****A. INNER JOIN****

The INNER JOIN keyword selects all rows from both the tables as long as the condition is satisfied. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be the same.

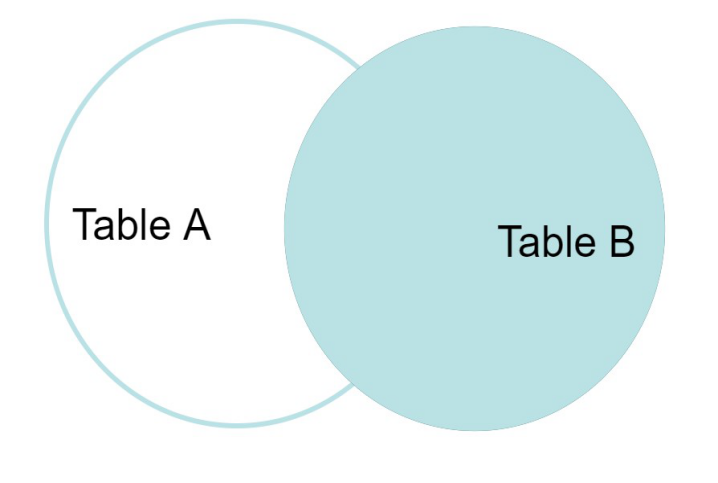


### ****B. LEFT JOIN****

This join returns all the rows of the table on the left side of the join and matches rows for the table on the right side of the join. For the rows for which there is no matching row on the right side, the result-set will contain *null*. LEFT JOIN is also known as LEFT OUTER JOIN.



### ****C. RIGHT JOIN****

RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of the join. For the rows for which there is no matching row on the left side, the result-set will contain *null*. RIGHT JOIN is also known as RIGHT OUTER JOIN. 

### ****D. FULL JOIN****

FULL JOIN creates the result-set by combining results of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both tables. For the rows for which there is no matching, the result-set will contain *NULL* values.



* **Explain working methodology of agile model and also Write pros and cons.**

 Agile Methods break the product into small incremental builds.  These builds are provided in iterations.

**Pros:** 

1. Is a very realistic approach to software development
2. Promotes teamwork and cross training.
3. Functionality can be developed rapidly and demonstrated.
4. Resource requirements are minimum.
5. Delivers early partial working solutions.
6. Good model for environments that change steadily.
7. Good model for environments that change steadily.
8. Little or no planning required.
9. Easy to manage
10. Gives flexibility to developers

**Cons:**

* Not suitable for handling complex dependencies.
* Software Testing Assignment 
* 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.
* An overall plan, an agile leader and agile PM practice is a must without which it will not to meet the deadlines.
* There is very high individual dependency, since there is minimum documentation generated.
* **Draw usecase on online shopping product using COD.**

Open browser

Search product

Place order

select product

Select size

Seacrh quality

Add to cart(bag)

Sign in/enter address

exit

Payment option (cod)

* **Draw use case on online shopping product using**

**Payment gateway.**

Open browser

Place order

Payment option

Sign in/registration

Enter address

select option

Select quality

Select item

Add to cart(bag)