**Summary**

**Testing**:-

The main object is to verify the functionality of the given build whether it is working fine or not.

**SDLC life Cycle**: -

It is software development life cycle. In this there are 6 stages. According to these stages a software should be developed.

1.Requirement phase

2.Designing phase

3.Coding phase

4.Testing phase

5.Deployement

6.Maintanance

1.**Requirement phase: -**

In this phase Business Analyst will gather the all requirements from the client. According to that requirement BA will make BRS/SRS document.

**2.** **Designing phase**: -- ->

a. high level design

b. low level design

According to the BRS document software architect will design the software architecture

3.**coding phase**:-

In this phase the developers will develop the code according to the software architecture

4.**Testing phase** :-

In this phase tester will test the build whether it is working expected or not once the build is moved from dev environment to testing environment . If is there is any difference is happened between actual and expected then it is treated as bug/defect.

5.**Deployement phase :-**

In this phase if the build is ready in QA environment then it is moved to Preprod environment i:.e beta environment . If there is no bugs then it is moved production environment.

6. **Maintenance phase** :-

After the build is delivered to the client the company will give the maintenance service to some period of extent.

**Types of testing**: -

1.functional testing

2.Non-functional testing

1.**Functional testin**g: -

**Smoke testing**: -

Test the basic and critical functionality of the build

**Regression testing**: -

Test the existing functionality of the given build after adding the new features. If is there any impact on existing features depends upon the impact go for unit or full or regional regression.

**Retesting**: -

Test the last failed testcases. It is nothing but test the test cases after fixing the bug

**Sanity testing**: -

Test the stable build working fine or not. It is the subset of regression testing

**Exploratory testing**: -

Here the test cases are not prepared in advance . Testing done parallelly with exploring , designing and execution.

**Ad-hoc testing**: -

Here testing done randomly. First understand the requirements then test the software randomly.

**UAT testing** (user acceptance testing): -

It is done by client-side testers. Here testers verify that the build is working fine and it is working as expected or not

**System testing**: -

In system testing we need to the software end-end. That is each and every functionality is verified working as expected or not.

Integration testing: -

Here verifies the data flow between the modules or not.

2**.Non-functional testing: -**

**Compatibility testing :-**

Heretest the software in different modules and different environments.

**Usability testing** : -

Here the application is easily understandable to user or not nothing but whether software is user friendly or not.

**Security testing : -**

Heretest the software which protects data and working as intended or not.

**JAVA**

|  |  |
| --- | --- |
| Java basics | [https://www.udemy.com/course/java-programming-basics/](https://www.udemy.com/course/java-programming-basics/%20) |

Concepts involved in the above link:

**Java** is a platform independent, and it is object-oriented programming language.

Java programs can run on multiple platforms without re-writing the code in separate platforms.

Here we are creating object for non-primitive data types that’s why it is object oriented programming language.

There are 3 steps involved in program execution.

1.**source** **code**: - code which is written by developers

2.**Compilation**: - compiler will convert the source code into .java class file .

3. **Execution**: - JVM (Java virtual mem) will convert the .java file into byte code. JVM will execute the program.

**Data types**: -

There are 2 types of data s are used.

1.Primitive datatypes (System define)

2.Non primitive datatypes (User Define)

Primitives are int, char, long, byte, short, Boolean, float, double.

Non primitives are Arrays, Strings, objects and classes.

Variables**: -**

It is a container used to store the values.

Ex: -

Int a; -----🡪1

a=20. ----🡪2

a=30; -----🡪3

Here int -datatype

A=variable

20=value

From 1-----------------------------------🡪variable declaration

From 2---Here am assigning the value to a ---🡪Variable Initialization

From 3---we can re assign the value into the same variable which is----------🡪variable Re-initialization.

There are 2 types of variables are there .Those are Local and Global

**Local variables**: **- The name itself indicates these are belongs to that particular method onl**y. These are not used outside of the method. There are 2 types.

1.**Static** – which belongs to the class only. We can directly access these variables in both static and non-static methods.

2.**non-Static** – These are not belonging to class. So, we need to create object for calling these methods. We can directly call these variables in non-static methods, but it needs object creation for calling into statical methods.

**Global variables: -** These variables are represent in inside the class and outside the method. **These** variables are used anywhere in the class.

Operators**: -**

There are different types of operators are in java.

1.Arthematic operators 🡪 (+, -, \*, %) – perform arithmetic operations

2.Logical operators -🡪 (&&, ||, !) –perform logical operations

3.conditional operators (true, false)

4.Relational operators (>, <, >=, <=) -perform comparison

By using of these can perform multiple operations.

Methods: -

Method is a set of or block of instructions to perform a code.

It is a way of performing some operation.

Methods are either static or non-static.

Ex: - public static void method () {

} -----------------------------🡪 static method

Public void method () {

} -------------------------------🡪 non-static method.

In above methods we directly take the data in the method.

Sometimes we can take data as parameters. such as parameterized or parameters passing methods.

Ex: - public static void method (int a, int b) {

}

Return types: -

Generally, if the method contains void, it doesn’t return any type of data.

If we are replacing any data type instead of void then the method returns that value.

Ex: - public int method (int a) {

Return a; ----------------------🡪 it returns int value

}

**Main method**: -

Public static void main (String [] args ){

}

Here

Public: -Access specifier- it gives the information about how to access

Static: - Type of a method

Void: - it doesn’t return anything

Main -it indicates that main method.

String [] args—string array arguments.

JVM will consider this as a main method and it executes the program from main method.

JVM will checks the one by one lines in main method it will executes the all the methods.

Conditional Statements**: -**

Thereare 3 types of conditional statements.Those are

1.if statement

2.if-else if

3.Nested if

1.**If statement**: -

If the condition satisfies then it executes if statement .

If the condition is not satisfying it executes else statement.

Int a=30

If(a>=20) { --------------🡪 if its ok execute if statement

} else {

} -----------------🡪 if not executes else statement

2.**if-else if**: -

If the condition is satisfied then it executes if statement.

If the condition is not satisfied it verifies the else if blocks.

If any of the above conditions are not satisfied then it executes else block.

Int a=69

If(a==64) {

} else if(a>69) { ---------🡪 here condition not satisfies it goes to else block.

} else{

}

**3.Nested if: -**

Here the condition is not satisfied it directly goes to else block.

If condition in if satisfies, then it goes to nested if.

Int a=65

If(a>56){

If(a<98){

} else {

}

}

else {

}

If nested if statement is not satisfied it executes the else part of that if condition.

Loops: -

Loops are used to iterate the same operation many times.

Always check the condition before perform the operation.

Loops in java are

1.For loop

2.while loop

3.do-while

4.for-each loop

**For loop**: -

Int a=15

For(int i=0;i<=a; i++/i--){

}

**While** loop:-

Int a=18;

While(int i<a){

i++/i--

}

**Do-while**: -

It is quite different than while and for. In while and for first check the condition and then go for operation.

Whereas in do while first perform the operation then go for condition

int a=3

do {

System.out.println(a);

a++

} while(a>=3)

These are the commonly used loops in java coding.

**Classes and Objects:** -

Class is nothing but a blueprint. It contains the all the methods , variables and behaviour of the objects

**Object: -**

It is a real time entity.

Object can be created with the help **new** keyword.

It should be mandatory for calling non static methos and variables.

In a single class we can create multiple objects.

**Access Modifiers**: -

Access modifiers are also called as access specifiers. It specifies the accessibility of class, methods etc.

4 types of access specifiers: -

1.Public ---we can access anywhere in the project

Public void method ()

2.Private----we can access with I that class only. Not to outside of class and package.

Private void method ()

3.Protected---we can access anywhere in the class .And it can access inside the package as well with the help of inheritance.

Protected void method ()

4.Default----we can access with in the class. It should not allow outside of the package. Here we cant use any access modifier while writing the signature.

Void method().

Packages :-It is collection of classes

Inheritance: -

Acquiring the properties from parent class to child class.

It is the main concept of JAVA OOPS concept.

By using extends keyword we can inherit the classes.

Only non-static methods should be inherited.

There are 5 types of inheritances :-

1.single level-only single child class will be inherited

2.Multilevel -no of child classes should be inherited from the parent class.

3.hierarchical- single parent, multiple child classes

These 3 types can be achieved through classes.

4.Multiple-single child and multiple parent classes.

5.Hybrid-combination of both multiple and hierarchical inheritance

These 2 types can’t be achieved through classes due to **AMBIGUITY ERROR.** So we can achieve through interfaces.

**Overriding: -**

The both parent and child class having a same method name and variables and Same method signature is method overriding.

We can override non-static methods only.

It is important to achieve run time polymorphism.

Abstract Classes and Methods: -

The classes which are having in complete method bodies are called abstract classes.

We cant create object for abstract methods.

If the class having all the static methods, then that class is a abstract class.

**Abstraction:-**

It means showing the functionality and hiding the implementation to the end users is known as Abstraction.