JAVA

Java is an object-oriented programming language. Everything in Java is an object which means to organize the software as a combination of different types of **objects** that incorporates both data and behavior.

Java is an Object Oriented language. The benefits of using java as programming language are portable, secured and platform independent.

JVM - Java Virtual Machine

(Java Virtual Machine) provides a runtime environment in which Java bytecode can be executed. It can also run those programs which are written in other languages and compiled to Java bytecode. At compile time, java file is compiled by Java Compiler (It does not interact with OS) and converts the java code into bytecode.

JRE - Java Runtime Environment

JRE (Java Runtime Environment) is a set of software tools which are used for developing Java applications. It is used to provide the runtime environment. It is the implementation of JVM. It contains a set of libraries + other files that JVM uses at runtime.

JDK - Java Development Kit

JDK (Java Development Kit) is a software development environment which is used to develop Java applications and applets. It physically exists. It contains JRE + development tools.

Naming Conventions

Java is case sensitive language. The naming conventions to be followed while writing a java codes are…..

Package: The first letter name should be in lowercase letter. Ex: selenium	**Class:** The every first letter of the class name should be in Upper case. (PascalCase) Ex: SampleCode
Method: The method name starts with lowercase and should follow the camelCase Ex: applyBreak().	**Variable:** The variable declaration should be with camelCase letter. int wheelsCount=4;

Note: Java program file name should exactly match with class name and should be same as any predefined classes like String, Array.

Data Types
Primitive Data Types
i) Integer Types: 1) byte (8 bits); Example: byte a =10; 2) short (16 bits); Example: short a =1000; 3) Integer (32 bits); Example: int i = 10000; 4) long (64 bits); Example: long l =100000000000L;
ii) Relational types (Numbers with decimal places): 1) float (32 bits); Example: float f = 1.23f or (float) 1.23; 2) double (64 bits); Example: double d =123.4567890;
ii) Characters: 1) Character (16 bits); Example: char c ='Z' iv) Conditional 2) Boolean (1 bit); Example: boolean b = true;
Non-Primitive Data Types
Non-primitive or Reference data types in Java are Objects, Class, Interface, String and Arrays. Example: String str = "Java World";

Variables in Java	
a) Local variable: Local variable is declared within methods or blocks.	**a) Instance variable:** Instance variables are declared inside the class but outside the body of the method. It is called instance variable because its value is instance specific and is not shared among instances.

Example:

```
public class Test{
 int i1=10; //instance variable
public void testMethod(){
 int i3=30; //local variable
 } } ;
```

Operator	
Arithmetic Operators: 1) Addition + (for Addition, String concatenation) 2) Subtraction – (for Subtraction) 3) Multiplication * 4) Division / 5) Modules % 6) Increment ++ 7) Decrement —	**Relational Operators:** 1) == comparison (equals) 2) ! = Not equal 3) > greater than 4) >=greaterthan equal to 5) < lessthan 6) <=lessthan equal to
Assignment Operators: 1) Assignment Operator: = 2) Add and Assign: += 3) Subtract and assign: -= 4) Multiple and assign: *=	**Logical Operators:** 1) Logical Not Operator: - ! 2) Logical And Operator: && → executes the loop when all the given conditions are true. 3) Logical Or Operator:|| ->executes the loop with any one condition to be true.

Control Statements	
Looping Statements	

for loop
syntax : for(initialization ;condition; increment/decrement) {
// code to be executed }	

while loop while(condition){ // code to be executed }	

for –each loop
for (type var : array) {
// code to be executed}
	do-while loop
	do
	{code to be executed} while(condition)

Jump Statement	
Break: When Condition satisfied then stops the execution of the loop and continues with the next block of code	**Continue:** When Condition satisfied then skips the current iteration and the next block of code

Decision making Statements	

If
If(condition) { code to be executed }
Example if(a<b){
System.out.println("b is bigger number"};
	If-else
	If(condition)
	{code to be executed }
	else{code to be executed};

Nested **If** If(condition) {if(condition){ code to be executed }}	**If-else-if** If(condition) {code to be executed }; else if{code to be executed}; else if{code to be executed}; else{Code to be Executed};
Switch case Switch(expression) { case exp1: Statement 1; break; case exp2: Statement 2; break; default default statement; break; }	

Java Objects

Objects –represents the state and behavior of the class variable and methods

Syntax to call class variables and methods:

ClassName obj=new ClassName();

Java Execution starts with main method:

Main method signature is as **public static void main(String [ ] args)**

Sample code:

```
public class LearnVariables{
{
int b = 15;
public static void main(String[] args) {
LearnVariables var= new LearnVariables();
System.out.println(var.b);
}
}
```

Access Specifiers
- explain the scope/visibility of class variables and methods of a Java Program
Public: accessible to all class (Global)
Private: accessible only to the current class object
Protected: accessible to the current class, subclasses and the packages
Default: accessible only to class and the package ***Packages:** includes the group of classes/subclasses