Selenium 4 - Overview of Java Programming

Java has three important editions

i) Java Standard Edition / Core Java (Old name J2SE)

ii) Java Enterprise Edition / Advanced Java (Old name J2EE)

iii) Java Micro Edition (Old name J2ME)

Java Standard Edition or Core Java is enough for Automated Testing with Selenium,

----

Note 1: In Computer Programming 80% concepts are common (in every programming language), Syntax may vary from one language to another

Example: Data Types, Variables, Operators, Control Flow (Conditional, Loop and Branching) etc...

Note 2: Some features may vary from one language to another, Example: Functions in C Languages, Methods in Java....

Overview of Java Standard Edition / Core Java:

1) Comments

- To make the code readable
- To make the code disable from execution

Note: Java supports Single line and multiple line comments

2) Data Types:

• A Data type is a classification of the type of data that a variable or object can hold in Computer programming

Examples:

Character,

Integer,

Float, Double....

String

Boolean etc...

Java supports two categories of Data Types,

i) Primitive Data Types

ii) Non Primitive Data Types / Reference Data Types

3) Java Modifiers

Modifiers are used to set access levels for classes, variables and methods,

Java supports two categories of Modifies

i) Access Modifiers (default, public, private, and protected)

ii) Non Access Modifiers (static, final, abstract,)

4) Java Variables

A named memory location to store or hold temporary data within a program

(RAM

ROM - HDD, CD, DVD, USB Drive etc...)

Java Supports three types of variables,

i) Local Variables

ii) Instance Variables

iii) Static Variables/Class Variables

Note: Java supports explicit declaration of variables

In Java:

```
int a;
a=100;//Correct
b=200; //Incorrect
int c=a+b; //Incorrect
```

----

int a;

int a, b, c;

int d=100;

int e=200, f=300, g=400;

In VBScript:

Dim a

a=100 'Correct (Explicit variable)

b=200 'Correct (Implicit variable)

Msgbox a+b '300

Note: VBScript supports Implicit and Explicit declaration of variables

5) Operators in Java

Operators are used to perform Mathematical, Comparison and Logical operations

Categories of Operators in Java,

i) Arithmetic Operators

ii) Assignment Operators

iii) Relational / Comparison Operators

iv) Logical Operators

----

v) Bitwise Operators

vi) Miscellaneous Operators

etc...

6) Java Control Flow Statements

i) Conditional / Decision Making statements (if, switch,)

ii) Loop Statements (for, while, do while and enhanced for)

iii) Branching statements (break, continue and return)

a) Java Conditional Statements / Decision Making Statements

i) Two Types of Conditional Statements

1) if statement

2) switch statement

ii) Three types of Conditions

1) Single Condition (Positive / Negative Condition)

```
if (a>b){
.
}
---------------
if (b<a){
}
----------------------------
```

2) Compound / Multi Condition

```
if ((a>b) && (a>c) && (a>d)){
.
}
```

3) Nested Condition

```
if (a>b){
  if (a>c){
     if (a>d){
     ...
    }
  }
}
```

iii) Usage of Conditional statements

1) Execute a block of statements when a condition is true

2) Execute a block of statements when a condition is true otherwise execute another block of statements

3) Execute a block of statements when a compound condition is true otherwise execute another block of statements

4) Decide among several alternates (else if)

5) Execute a block of statements when more than one condition is true (nested if)

6) Decide among several alternates (using switch statement)

b) Loop Statements

Used for repetitive execution

i) for loop

ii) while loop

iii) do while loop

iv) enhanced for loop (Array)

c) Branching Statements

- Branching statements are used to transfer control from one point to another in the code
- Branching statements are defined by three keywords - break, continue and return

7) String handling in Java

- Sequence of characters written in ""
- String may contain Alphabets, Numeric, Alpha-numeric and special characters.

Example:

"SELENIUM"

"selenium"

"India123"

"123"

123 //Integer

"India123*&^"

Etc...

Note: Using String predefined methods we can perform operations on strings

8) Arrays

- Generally, Array is a collection of similar type of elements

Dim a, b(3)

b(0)=100

b(1) ="India"

b(2) =10.234

- In Java Array is an Object that contains elements of similar data types
- Java Array is index bases and index starts from zero
- The length of an Array is established when they are is created and Array length is fixed,
- Each item in an Array is called as Element

9) IO Operations (includes File Handling)

- Read Data using Input devices
- Display Output on the Console
- Read data from files, Write data to files

10) Methods

What is Method?

A Set of statements to perform an Operation

Why Methods?

Methods are used for code reusability

When we choose methods?

Whenever we want to perform any operation multiple times then we choose methods

Note: All programming features can be used in methods also

Two Types of Methods in Java,

i) Built-in Methods

Number Methods

Character Methods

String Methods

Array Methods Etc...

ii) User Defined Methods

Method with returns a value (Static and Non Static methods)

Method with returns nothing (Static and Non Static methods)

11) Exception Handling in Java

- In Computer programming exception is an abnormal condition,
- An exception is an event that occurs during the execution of a program that disturbs the nor flow of instructions
- The exception handling in Java is one of the powerful mechanisms to handle run-time errors

int a=10; //No Error

int b=10 //Syntax Error

int c= 10/2; //No Error

int d=10/0; //Run-time Error (* Exception handling is required)

12) Java Object Oriented Programming (OOPS) Fundamentals

i) Inheritance

ii) Polymorphism

iii) Abstraction

iv) Encapsulation

i) Inheritance

- Inheritance is a mechanism in which one object acquires all the properties and behaviors of parent object
- Using Inheritance we can create classes that are built-in upon existing classes
- When we inherit from an existing class then we can reuse methods and fields from the parent class

----

Java Supports,

a) Single Inheritance

ClassB extends ClassA

b) Mluti Level Inheritance

ClassB extends ClassA

ClassC extends ClassB

c) Multiple Inheritance (* Java doesn’t support multiple Inheritance)

ClassB extends ClassA

ClassB extends ClassZ

ii) Polymorphism

Performing task/s in different ways,

Polymorphism derived from two Greek words,

Poly - Many

Morphs - Forms / ways

So Polymorphism means "Many Ways"

Two types of Polymorphism in Java,

a) Compile Time Polymorphism / Method OverLoading

b) Run-time Polymorphism / Method Overriding

iii) Abstraction

• Abstraction is a process of hiding implementation details and showing functionality to the user

• In another way, it shows important things to the user and hides internal details Ex: Sending Email

• Abstraction focuses on what the object does instead of how it does

Abstraction can be achieved in two ways,

1) Abstract Class

2) Interface

iv) Encapsulation

• Encapsulation is a process of wrapping code and data together into a single unit

Ex: Capsule

It provides control over the data

Java Programming

Java Project

Java Package/s

Java Class/s

We write Java code in Java Class...
