

Selenium 5 - Java Environment Setup

- 1) Uses of Java
- 2) Java Syntax
- 3) Java Environment Setup & Verify

1) Uses of Java

Java is used to develop

- Desktop Applications

- Web Applications

- Enterprise Applications (ex: Banking, Insurance, ERP etc...)

- Mobile Applications

- Embedded Systems

- Smart Cards

- Games Applications

- Scientific Applications Etc...

Test Automation (with Selenium)

2) Java Syntax Rules:

- i) Java is a case sensitive language

Note: All Java keywords and reserved words are small letters
(if, for, public, main, true, false, null...)

ii) First letter of class name should be in upper case

sample //Incorrect

Sample //Correct

Firstprogram //Correct

FirstProgram ////Correct

iii) Java Method names should start with lower case

iv) Java program file name should exactly match with class name

v) Java program execution starts from main method, which is mandatory in every Java program

vi) Every statement /step should end with semi colon (;)

vii) Code blocks (conditional, Loop, Method etc...Enclosed with {}),

viii) Java supports Explicit declaration of Data Types

In Java,

int sno =123;//Correct

int x;//Correct

char a='A'; //Correct

boolean y=true;

abc =100; //Incorrect

In VBScript

```
Dim city
```

```
city =100
```

```
.
```

```
city ="India"
```

```
.
```

```
city=1.23
```

```
.
```

```
city=#10/10/2010#
```

```
-----
```

ix) Java supports explicit declaration of Variables

```
int a, b;
```

```
a=10;
```

```
b=20;
```

```
c=30;//Incorrect
```

In VBScript

```
Dim a
```

```
a=100
```

```
b=200 'Correct
```

```
-----
```

3) Java Environment Setup

To write and execute Java programs then Java Environment is required

Three important steps in Computer programming,

i) Write a Program (in any Editor)

- ii) Compile the Program
- iii) Run the program

A) Steps for writing and executing Java programs using command line interface

- i) Download Java Software (JDK) and Install
- ii) Set Java Environment variable in the OS Environment
- iii) Write a Java program in Notepad and save with ".java" extension
- iv) Compile the java program file in command prompt and Run...

B) Steps for writing and executing Java programs using Eclipse IDE

- i) Download Java Software (JDK) and Install
- ii) Set Java Environment variable in the OS Environment
- iii) Download Eclipse IDE and Extract
- iv) Write Java programs in Eclipse Editor and Run

Download Java Software (JDK) from either java.com or oracle.com, and install

Note: Download Java Software based on your Operating Environment

Ex: Windows 10 32 bit OS

* After Java Software installation we get "Java" Folder in C:/Program Files/

Set Java Environment Variable...

Copy jdk bin directory path from your computer,

Navigation to set Java Environment Variable:

Select MyComputer and Right Click

- > Properties
- > Click "Advanced System Settings"
- > Click "Environment Variables"
- > Select "Path" Variable in System Variables pane
- > Click "Edit"
- > Paste Jdk bin directory path
- > Click "OK"
- > Click "OK"
- > Click "OK"

Step 1: Write a Program in Notepad

```
public class Sample{  
    public static void main (String [] args){  
        System.out.println("Hello Java");  
    }  
}
```

save as .java file with class name

Step 2: Compile the Program

- > Launch the Command prompt
- > Change to Java program file directory
- > Type javac File Name/sample.java

Then it will create Java class file (.class)

Step 3: Run the program

- > Launch the Command prompt
- > Change to Java program file directory
- > Type Java Class file name without extension

It will display the output on the Console

Eclipse IDE:

> Eclipse IDE is a platform to write & execute Computer programs like Java, C, C++, Perl, Python, Ruby, PHP etc...

> Eclipse IDE is Open Source

> It provides Editor for writing Programs, Syntax Guidance, Context Help and Auto compilation etc...

Navigation for writing and executing Java programs in Eclipse IDE

> Launch Eclipse IDE

- > Create Java project
- > Create Java Package under the Java Project
- > Create Java Class under the Java Package

Write Java code in the Java Class...

Java Program Example:

```
public class Sample {  
  
    public static void main(String[] args) {  
        int a=10, b=20;  
        System.out.println("Addition of a, b is: "+ (a+b));  
  
        if (a>b){  
            System.out.println("A is a Big Number");  
        }  
        else {  
            System.out.println("B is a Big Number");  
        }  
    }  
}
```
