Introduction:

Selenium is an open source automation testing tool.
It is used exclusively for web based applications
You can work on multiple operating systems using selenium

Platforms Supported by Selenium

WINDOWS

OS X(MAC)

LINUX

SOLARIS

Selenium Browsers Support:

INTERNET EXPLORER

FIREFOX

CHROME

SAFARI

WE WILL BE USING SELENIUM VERSION 3.0+

WHAT IS JAVA, AND WHY IT IS USEFUL?

Java is a programming language in the tradition of C and C++. As a result, if you have any experience with C or C++, you'll find yourself in familiar territory often as you learn the various features of Java. However, Java differs from other programming languages in a couple of significant ways. The following sections describe the most important differences.

PLATFORM INDEPENDENCE

One of the main reasons Java is so popular is its *platform independence,* which means that Java programs can be run on many different types of computers. A Java program runs on any computer with a *Java Runtime Environment,* also known as a *JRE,* installed. A JRE is available for almost every type of computer — PCs running Windows, Macintosh computers, Unix or Linux computers, huge mainframe computers, and even cell phones.

OBJECT ORIENTATION

Java is inherently *object-oriented,* which means that Java programs are made up of programming elements called *objects.* Simply put, an object is a programming entity that represents either some real-world object or an abstract concept.

All objects have two basic characteristics:

- Objects have data. For example, an object that represents a book has data such as the book's title, author, and publisher.
- Objects also have *behavior,* which means that they can perform certain tasks. In Java, these tasks are called *methods*. For example, an object that represents a car might have methods such as start, stop, drive, or crash. Some methods simply allow you to access the object's data.

CLASSES IN JAVA:

- A CLASS IS A BLUEPRINT FROM WHICH INDIVIDUAL OBJECTS ARE CREATED.

  A SAMPLE OF A CLASS IS GIVEN BELOW:

```
public class Dog {
  do something…
}
```

- VARIABLE IS A PLACEHOLDER WHERE YOU CAN STORE SOME INFORMATION. THINK OF A VARIABLE AS A NAME AND NAME HAS A VALUE. FOR INSTANCE,

  username = John

- WHEN WRITING A STATEMENT IN JAVA CLASS, EACH STATEMENT MUST END WITH A **SEMICOLON ;**

public static void main(String[] args) class

A Java **method is a collection of statements that are grouped together to perform an operation

- Java main method is the entry point of any java program
- Java main method is the only method in java that allows you to execute your codes/statements. Its syntax is always **public static void main(String[] args){ }**
- Also, String array argument can be written as **String[] args** or **String args[]**
- As you can see below that classes and methods always opens with **{** and closes with **}**

Test.java

```
1  package com.journaldev.java.examples1;
2
3  public class Test {
4
5      public static void main(String args[]) {
6
7          System.out.println("Hello World");
8
9      }
10 }
```

- If we want to print some information, we can use following command **System.out.println()** which allows you to print after main method is executed and it will appear in your log screen

String & int datatype

In Java we can define number & values in two different ways.

- **String** is anything that is defined inside double quotations " " by storing a value into a variable. For instance, alphanumeric, alphabet, special characters & even numeric, etc…

  String variable is defined as followed

  String myVariable = "Welcome to Automation class";

- Number is defined by keyword **int** and it can't be defined with quotations. For instance,

  int myNumber = 100;

- Now if I want to print out both variables, I can use **concatenation(+)** with print command
- Concatenation is used when two or more variables need to be combined either to print or to perform an operation

For instance, **System.out.println("My values are " + myVariable + " " + myNumber);**