Index

Installing JAVA

Go to URL : https://www.oracle.com/technetwork/java/javase/downloads/index.html
Click on : Java Platform (JDK)
Choose from below downloads a per your PC & OS configuration.
For 32 bit OS :- Choose “Windows x86”
For 64 bit OS :- Choose “Windows x64”
Install Downloaded JAVA.

How to Set Path in JAVA

1. Go to My Computer -> Properties -> Advanced system settings -> Environment Variables -> New tab of user variables -> Write path in variable name -> Write path of jdk bin folder in variable value -> ok -> ok -> ok

Or

2. Go to My Computer -> Properties -> Advanced system settings -> Environment Variables -> System variables -> Double click on option ‘path’ -> Go to the end of path value -> insert semicolon(;) -> Write path of jdk bin folder in variable value -> Again insert semicolon(;) -> Write path of jre bin folder -> Click ok -> ok -> ok

How to check JAVA is successfully installed or not

1. Go to “C:\Program Files”, In this location look for folder called “Java”. If this folder is present that means JAVA is successfully installed in your system.

And

2. Go to Command Prompt and type “java -version”, Installed Java version should display.

Installing Eclipse

Download from URL : https://www.eclipse.org/downloads/packages/
Version :- Eclipse IDE for Java EE Developers

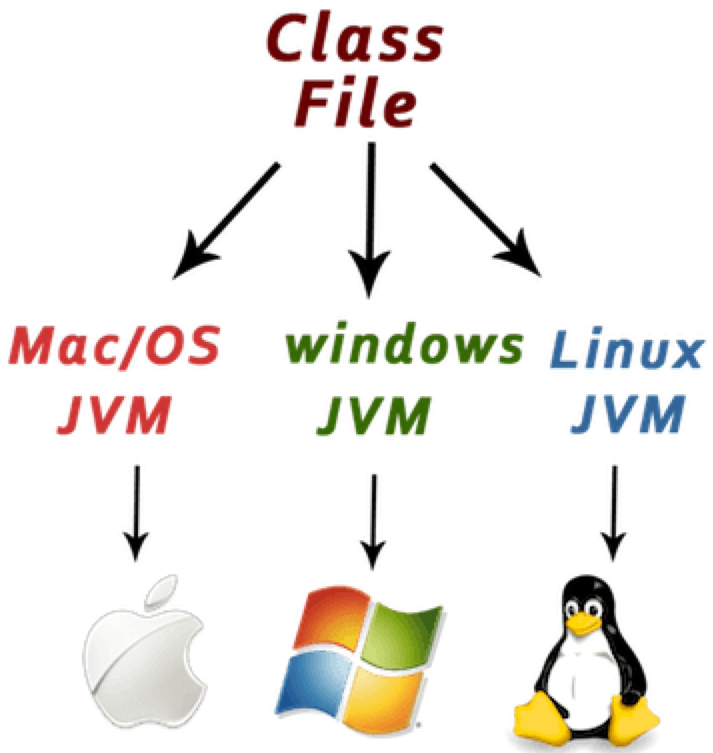
Features of JAVA

1. **Object-Oriented** - Java is an object oriented programming language. Object-oriented programming (OOPs) is a methodology that simplifies software development and maintenance by providing some rules.

Basic Concepts of OOPs are:

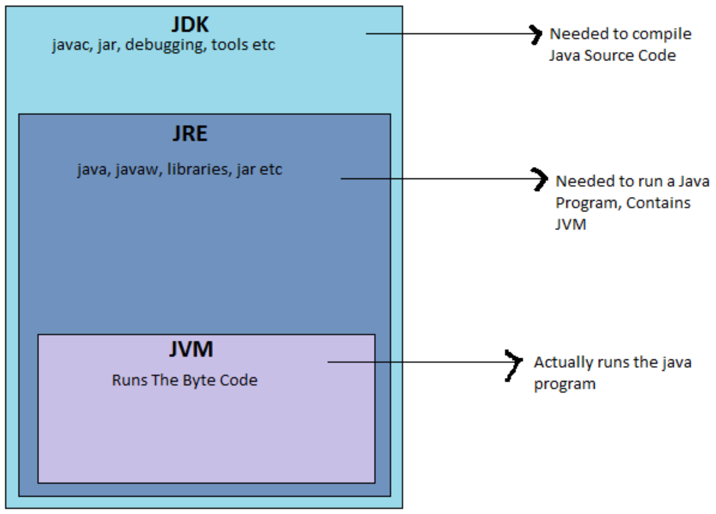
- Object
- Class
- Inheritance
- Polymorphism
- Abstraction
- Encapsulation

2. **Platform Independent** - Java code can be run on multiple platforms, for example, Windows, Linux, Sun Solaris, Mac/OS, etc. Java code is compiled by the compiler and converted into bytecode. This bytecode is a platform independent code because it can be run on multiple platforms, i.e., Write Once and Run Anywhere(WORA).



3. **Multithreaded** - A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area. Threads are important for multi-media, Web applications, etc.

JDK, JRE & JVM


JDK
javac, jar, debugging, tools etc
Needed to compile Java Source Code
JRE
java, javaw, libraries, jar etc
Needed to run a Java Program, Contains JVM
JVM
Runs The Byte Code
Actually runs the java program

First JAVA Program (Shortcut to run the java program - CTRL + F11)

```java
public class First_Java_Program {

	//function – Double Slash Used to comment the line
	public static void main(String[] args) { // Main Method
	System.out.println("Hello World");

//Shortcut for writing "System.out.println" :- write 'syso' and press cntrl+space
	System.out.println(12345);
	System.out.println("12345");
	}

}
```

Concept of class file

1. Class file in java is generated when you compile your program (.java file) using any java compiler like Sun’s javac which comes along JDK installation and can be found in JAVA_HOME/bin directory.

2. Platform Independent : Class file contains byte codes. Byte codes are special platform independent instruction for Java virtual machine.
Byte code is not a machine language. Since every machine or processor e.g. Intel or AMD processor may have different instructions for doing same thing, it’s left on JVM translate byte code into machine instruction and by this way java achieves platform independence.

Data Types

Data Types are of two types **"Primitive & Non-Primitive".**

Primitive :- boolean, char, byte, short, int, long, float, double

Non-Primitive :- String, array

Value and range of these data types are given below :-

Type	Size (bits)	Min Value	Max Value	Default Value
byte	8 bits = 1Byte	-128	127	0
char	16	0	65535	\u0000
short	16	2^15	2^{15} -1	0
Int	32	-2^31	2^{31} -1	0
Long	64	-2^63	2^{63} -1	0L
Float	32	2^{-149}	$(2-2^{-23}) . 2^{127}$	0.0f
Double	64	2^{-1074}	$(2-2^{-52}) . 2^{1023}$	0.0d
Boolean	1	--	--	FALSE

Example :-

```
public class DataTypes {
	// function
	public static void main(String[] args) { // Main Method
		int x = 100; // Can not store decimals - size of int is "32 bits"
		System.out.println(x);
		x = 200; // No need to declare "x" again
		System.out.println(x);

		long var = 3000L; // can not store decimals - 64 bits in size
		System.out.println(var);

		char c = 'y'; // single inverted comma
		System.out.println(c);

		boolean b = true; // true or false
		System.out.println(b);
		System.out.println(3 > 12);

		int q = 100;
		int w = 200;
		boolean x1 = q > w;
		System.out.println(x1);
	}

}
```

String Class

1. String is a class in Java.
2. These are basically sequence of characters.
3. It should be given in double quotes (“Hello”).
4. Spaces given inside double quotes are also considered as part of string.
5. Example :- String str = “Hello World”;

How to calculate length of string?

```
String str = "Hello World";
            int len;
            len = str.length();
            System.out.println(len);
```

Output :- 11

If & nested If statements

1. If statement is a conditional branch statement.
2. It tells your program to execute a certain section of code only if a particular test evaluates to true.
3. If boolean expression evaluates to true, the statements in the block following the if statement are executed.
4. If it evaluates to false, the statements in the if block are not executed.

Example :-

```
public class If_Nested If_Else {
      // function
      public static void main(String[] args) { // Main Method
            int q = 200;
            int w = 200;

            // If Statements
            if (q < w) {
                  System.out.println("q is lesser");
            } else if (q == w) {
                  System.out.println("q is equal to w");
            } else {
                  System.out.println("q is greater");
            }
      }
}
```

Java main method - public static void main(String[] args)

1. Java main method is the entry point of any java program. Its syntax is always **"public static void main(String[] args)"**
2. Also String array argument can be written as "**String… args** or **String args[]**"

3. **public** - This is the access specifier of the main method. It has to be public so that java runtime can execute this method.
#. What happens if we define the main method non-public as below :-
'static void main(String[] args)'
Output = Error (Main method not found in class 'class name')

4. **static** - When java runtime starts, there is no object of the class present. That's why the main method has to be static so that JVM can load the class into memory and call the main method. If the main method won't be static, JVM would not be able to call it because there is no object of the class is present.
#. What happens if we remove static from main method as below :-
'public void main(String[] args)'
Output = Error (Main method is not static in class 'class name')

5. **void** - Java main method doesn't return anything, that's why it's return type is void.
#. What happens if we have the main method as below :-
```
public static void main(String[] args){
return 0;
}
```
Output = Compilation Error (Void methods cannot return a value)

6. **main** - main(..) is the first method called by java environment. It's fixed and when we start a java program, Java runtime looks for the main method.
#. What happens if we have the main method as below :-
```
public static void mymain(String[] args){
}
```
Output = Error (Main method not found in class 'class name')

7. **String[] args** - The argument String indicates the argument type and arg is an array for string given during command line.
#. What happens if we have the main method as below :-
```
public static void main(String[]){
}
```
Output = Error (Main method not found in class 'class name')

System.out.println ("Hello Friends");
System is a predefined final class, **out** is a PrintStream object and **println** method is overloaded to print message to output destination, which is typically a console or file.