Java

By Mr. Ashok

Introduction to Java

- **Java is one of the world's most important and widely used computer languages, and it has held this distinction for many years.**
- **As of 2018, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers using and working on it.**

Creation Of Java

- **Java was developed by James Ghosling, Patrick Naughton, Mike Sheridan at Sun Microsystems Inc. in 1991. It took 18 months to develop the first working version.**
- **The initial name was Oak but it was renamed to Java in 1995 as OAK was a registered trademark of another Tech company.**
- **Java was initially launched as Java 1.0 and the current version of java is java 1.9.**

Application of Java

- Java is widely used in every corner of world and of human life. Java is not only used in software's but is also widely used in designing hardware controlling software components. There are more than 930 million JRE downloads each year and 3 billion mobile phones run java.
- Following are some other usage of Java :
  - Developing Desktop Applications
  - Web Applications like Linkedin.com, Snapdeal.com etc
  - Mobile Operating System like Android
  - Embedded Systems
  - Robotics and games etc.

Setup Java Environment

- For running Java programs in your system you will have to download and install **JDK kit from here**(current version is jdk 1.9).

Java Features

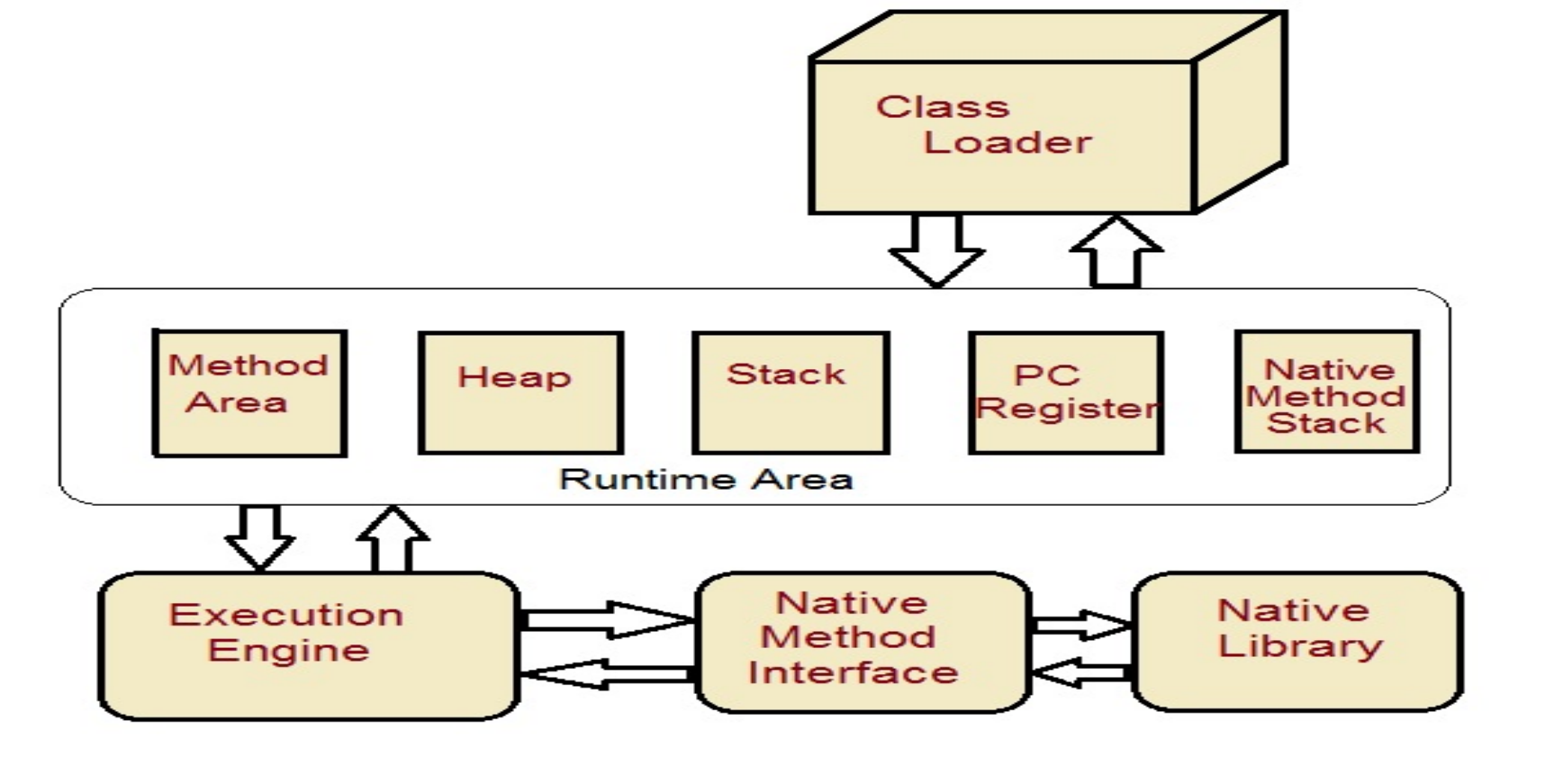
- **Simple**
- **Object Oriented**
- **Platform-Independent**
- **Robust**
- **Secure**
- **Multi-Threading**
- **Architectural Neutral**
- **Portable**
- **High-Performance**

Introduction to JVM

- **Java virtual Machine(JVM) is a virtual Machine that provides runtime environment to execute java byte code. The JVM doesn't understand Java typo, that's why you compile your *.java files to obtain *.class files that contain the bytecodes understandable by the JVM.**

- **JVM control execution of every Java program. It enables features such as automated exception handling, Garbage-collected heap.**

JVM Architecture

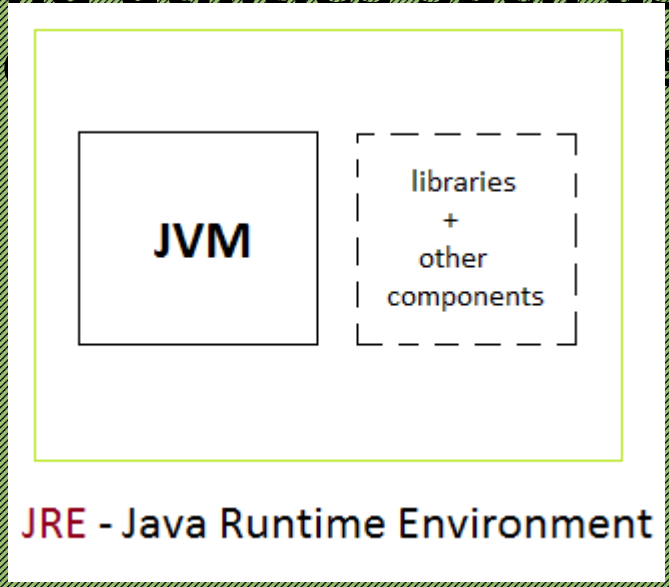


JVM Components

- **Class Loader : Class loader loads the Class for execution.**
- **Method area : Stores pre-class structure as constant pool.**
- **Heap : Heap is in which objects are allocated.**
- **Stack : Local variables and partial results are store here. Each thread has a private JVM stack created when the thread is created.**
- **Program register : Program register holds the address of JVM instruction currently being executed.**
- **Native method stack : It contains all native used in application.**
- **Executive Engine : Execution engine controls the execute of instructions contained in the methods of the classes.**
- **Native Method Interface : Native method interface gives an interface between java code and native code during execution.**
- **Native Method Libraries : Native Libraries consist of files required for the execution of native code.**

Differences between JDK JRE and JVM

- **JRE : The Java Runtime Environment (JRE) provides the libraries, the Java Virtual Machine, and other components to run applets and applications written in the Java programming language. JRE does not contain tools and utilities such as compilers or debuggers for [illegible]s and applications.**



Differences between JDK JRE and JVM

- **JDK : The JDK also called Java Development Kit is a superset of the JRE, and contains everything that is in the JRE, plus tools such as the compilers and debuggers necessary for developing applets and applications**

First Java Program

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

First Java Program-Explanation

- **class : class keyword is used to declare classes in Java**
- **public : It is an access specifier. Public means this function is visible to all.**
- **static : static is again a keyword used to make a function static. To execute a static function you do not have to create an Object of the class. The main() method here is called by JVM, without creating any object for class.**
- **void : It is the return type, meaning this function will not return anything.**
- **main : main() method is the most important method in a Java program. This is the method which is executed, hence all the logic must be inside the main() method. If a java class is not having a main() method, it causes compilation error.**
- **String[] args : This represents an array whose type is String and name is args. We will discuss more about array in Java Array section.**
- **System.out.println : This is used to print anything on the console like *printf* in C language**

Steps to Compile and Run your first Java program

Step 1: Open a text editor and write the code as above.

Step 2: Save the file as Hello.java

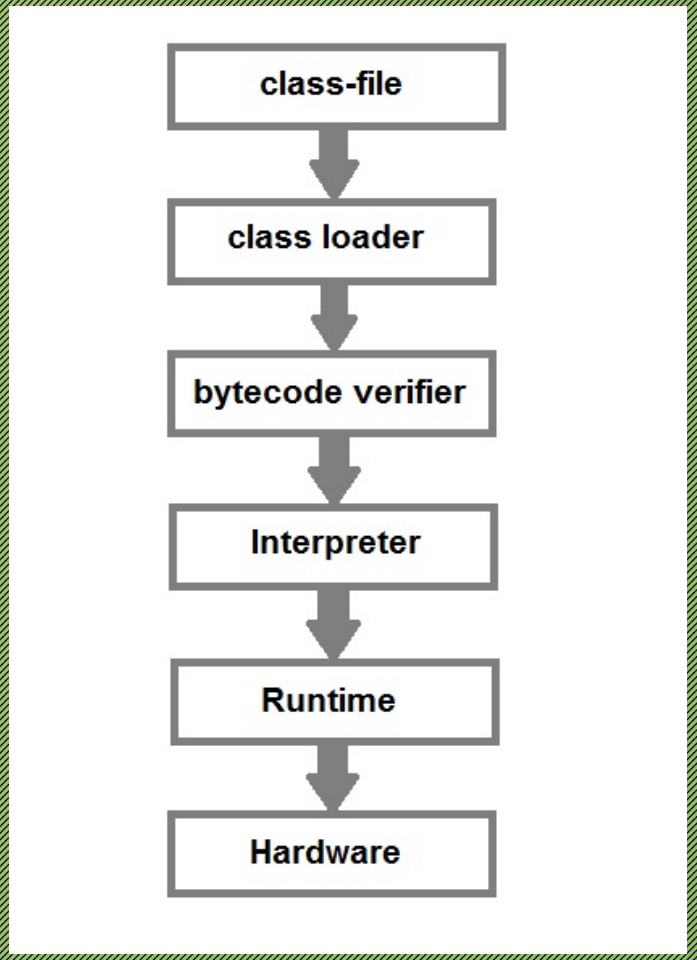
Step 3: Open command prompt and go to the directory where you saved your first java program assuming it is saved in C:

Step 4: Type javac Hello.java and press Return(Enter KEY) to compile your code. This command will call the Java Compiler asking it to compile the specified file. If there are no errors in the code the command prompt will take you to the next line.

Step 5: Now type java Hello on command prompt to run your program.

Step 6: You will be able to see Hello world program printed on your command prompt

What happens at Runtime



What happens at Runtime..contd

- After writing your Java program, when you will try to compile it. Compiler will perform some compilation operation on your program.
- Once it is compiled successfully byte code(.class file) is generated by the compiler.
- After compiling when you will try to run the byte code(.class file), the following steps are performed at runtime:-
- Class loader loads the java class. It is subsystem of JVM Java Virtual machine.
- Byte Code verifier checks the code fragments for illegal codes that can violate access right to the object.
- Interpreter reads the byte code stream and then executes the instructions, step by step.

Data Types in Java

- Java language has a rich implementation of data types. Data types specify size and the type of values that can be stored in an identifier.
- In java, data types are classified into two catagories :
  - Primitive Data type
  - Non-Primitive Data type

Primitive Data type

- A primitive data type can be of eight types :
- char
- boolean
- byte
- short
- int
- long
- float
- double

Integer group

This group includes byte, short, int, long

- **byte : It is 1 byte(8-bits) integer data type. Value range from -128 to 127. Default value zero. example: byte b=10;**
- **short : It is 2 bytes(16-bits) integer data type. Value range from -32768 to 32767. Default value zero. example: short s=11;**
- **int : It is 4 bytes(32-bits) integer data type. Value range from -2147483648 to 2147483647. Default value zero. example: int i=10;**
- **long : It is 8 bytes(64-bits) integer data type. Value range from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807. Default value zero. example: long l=100012;**

Floating-Point Number

This group includes float, double

- **float : It is 4 bytes(32-bits) float data type. Default value 0.0f. example: float ff=10.3f;**

- **double : It is 8 bytes(64-bits) float data type. Default value 0.0d.**

  example: double db=11.123;

Characters

- **This group represent char, which represent symbols in a character set, like letters and numbers.**

- **char : It is 2 bytes(16-bits) unsigned unicode character. Range 0 to 65,535.**

  example: char c='a';

Boolean

- **This group represent boolean, which is a special type for representing true/false values. They are defined constant of the language.**
  - **example: boolean b=true;**

Non-Primitive(Reference) Data type

- A reference data type is used to refer to an object. A reference variable is declare to be of specific and that type can never be change. We will talk a lot more about reference data type later in Classes and Object lesson.

Identifiers in Java

- **All Java components require names. Name used for classes, methods, interfaces and variables are called Identifier. Identifier must follow some rules. Here are the rules:**

- **All identifiers must start with either a letter(a to z or A to Z) or currency character($) or an underscore.**
- **After the first character, an identifier can have any combination of characters.**
- **A Java keyword cannot be used as an identifier.**
- **Identifiers in Java are case sensitive, foo and Foo are two different identifiers.**

Variables

- **When we want to store any information, we store it in an address of the computer. Instead of remembering the complex address where we have stored our information, we name that address. The naming of an address is known as variable. Variable is the name of memory location.**

- **Java Programming language defines mainly three kind of variables.**

  - **Instance variables**
  - **Static Variables**
  - **Local Variables**

Instance variables

- **Instance variables are variables that are declare inside a class but outside any method, constructor or block. Instance variable are also variable of object commonly known as field or property. They are referred as object variable. Each object has its own copy of each variable and thus, it doesn't effect the instance variable if one object changes the value of the variable.**

```
class Student
{
 String name;
 int age;
}
```

Static variables

- Static are class variables declared with static keyword. Static variables are initialized only once. Static variables are also used in declaring constant along with final keyword.

```
class Student {
    String name;
    int age;
    static int instituteCode=1101;
}
```

Additional points on static variables:

- static variable are also known as class variable.
- static means to remain constant.
- In Java, it means that it will be constant for all the instances created for that class.
- static variable need not be called from object.
- It is called by classname.static variable name
- Note: A static variable can never be defined inside a method i.e it can never be a local variable.

Local variables

- **Local variables are declared in method, constructor or block. Local variables are initialized when method, constructor or block start and will be destroyed once its end. Local variable reside in stack. Access modifiers are not used for local variable.**

```
float getDiscount(int price) {
      float discount;
 discount=price*(20/100);
 return discount;
 }
```

Arrays

- An array is a collection of similar data types. Array is a container object that hold values of homogenous type. It is also known as static data structure because size of an array must be specified at the time of its declaration.
- An array can be either primitive or reference type. It gets memory in heap area. Index of array starts from zero to size-1.
- It is always indexed. Index begins from 0.
- It is a collection of similar data types.
- It occupies a contiguous memory location.

Array – Program

```
class Test {
public static void main(String[] args) {
    int[] arr = {10, 20, 30, 40};
    for(int x : arr) {
    System.out.println(x);
    }
  }
}
```

String Handling

- **String is probably the most commonly used class in java library. String class is encapsulated under java.lang package. In java, every string that you create is actually an object of type String. One important thing to notice about string object is that string objects are immutable that means once a string object is created it cannot be altered.**
- **An object whose state cannot be changed after it is created is known as an Immutable object. String, Integer, Byte, Short, Float, Double and all other wrapper classes objects are immutable.**

Thankyou..!!