

## PRACTICAL - 01

AIM:- Study of JDK, JVM & JRE

THEORY:-

Java is a programming language & a platform. Java is a high level, robust, object oriented and secure programming language.

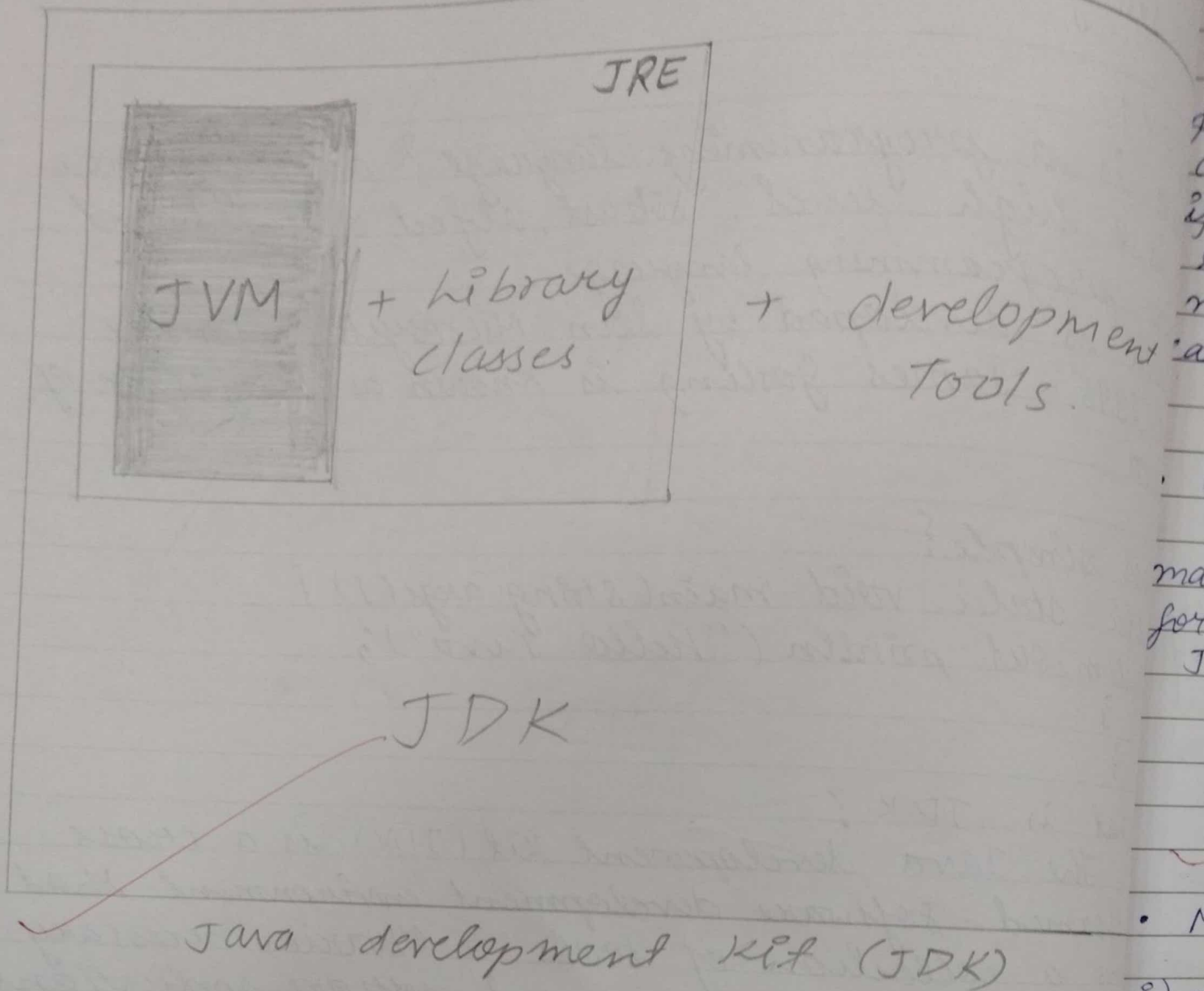
Java is developed by Sun Microsystems in the year 1995. James Gosling is known as the father of Java.

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

★) What is JDK?

The Java development kit (JDK) is a cross platformed software development environment that offers a collection of tools & libraries necessary for developing Java-based software applications and applets. It's a core package used in Java, along with the JVM (Java virtual Machine) and the JRE (Java Runtime Environment).

## Block diagram



- i) 0
- ii)
- iii)
- iv)

### • JRE VS JDK :-

If you are only interested in running java programs on your machine then you can easily do it using Java Runtime Environment. However, if you would like to develop a Java-based software application then along with JRE you may need some additional necessary tools, which is called JDK.

JDK = JRE + development tools.

### • Contents of JDK :-

The JDK has a private Java virtual machine (JVM) and a few other resources necessary for development of a Java application.

JDK contains:-

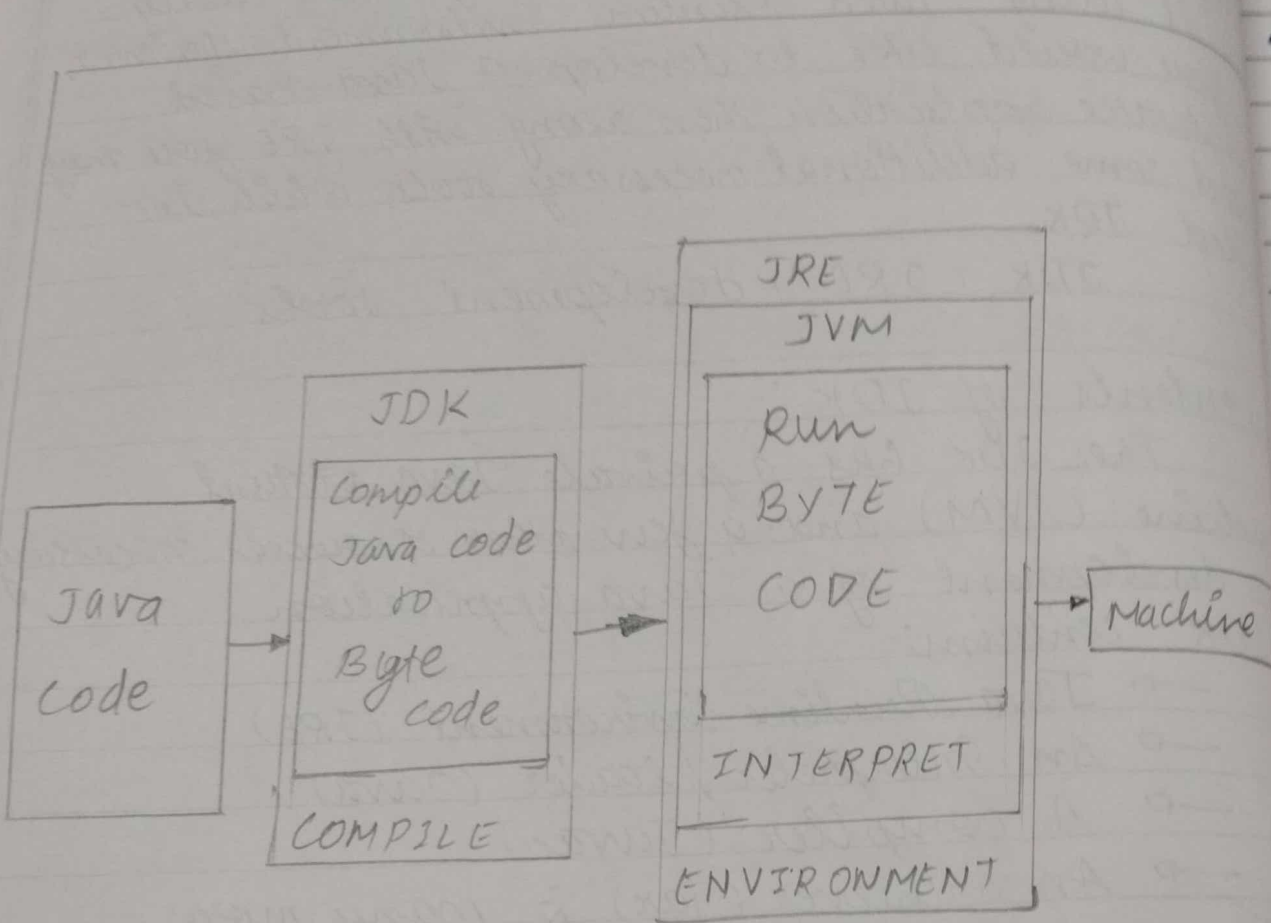
- Java Runtime Environment (JRE)
- An Interpreter/loader (Java)
- A compiler (Javac)
- An archiver (jar) & many more.

### • Most popular JDKs :-

- i) Oracle JDK :- The most popular JDK.
- ii) Open JDK.
- iii) Azul Systems Zing.
- iv) IBM J9 JDK.



## \* Block diagram.



## ARCHITECTURE

- Installation
  - 1) go to or the link
  - 2) select the and add it
  - 3) Just check on your co on :-

C:\pr

- Set JAVA

- 1) Right -
- 2) go to variables, the enac stored C:\P default

After Ins  
Comman  
this thr  
version  
restart

### • Installation of JDK:-

- 1) Go to Oracle's official download page through the link.
- 2) select the latest JDK version & click download and add it to your classpath.
- 3) Just check the JDK software is installed or not on your computer at the current location or:-

C:\program files\Java\jdk 11-0-9

### • Set JAVA - HOME for Windows :-

- 1) Right-click My computer & select properties.
- 2) Go to the advanced tab & select environment variables, & then edit JAVA - HOME to point to the exact location where your JDK software is stored.

C:\program files\Java\jdk 11-0-9 is the default location in windows.

After installing the JDK & JRE adds the Java command to your command line. You can verify this through the command prompt by the java version command. In some cases, you need to restart your system after installing JDK.

## Compile & Run Java code using JDK

You can use JDK compiler to convert your Java text file into an executable program. Your Java text segment is converted into byte code after compilation which carries the class extension.

First, create a Java text file & save it using a name. Here we are saving the file as `hello.java`.

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

After that, just simply use `javac` command, which is used for the compilation purpose in Java.

```
C:\program files\Java\jdk-11.0.9\bin\  
java.exe "Hello Java".
```

Now, you can run your code by using `java` command.

```
C:\Users\Documents > java hello-world
```

(Output) `hello world`



## \* JVM Architecture :-

JVM (Java Virtual Machine) acts as a run time engine to run java applications. JVM is the one that actually calls the main method present in a Java code. JVM is a part of JRE (Java Runtime Environment)

Java applications are called WORE (Write Once Run Everywhere). This means a programmer can develop java code on the one system & can expect it to run on any other Java-enabled system without any adjustment. This is all possible because of JVM.

When we compile a java file, class files (contains byte code) with the same class names present in java file are generated by the java compiler. This class file goes into various steps when we run it. These steps together describes the whole JVM.

### • class loader subsystem :-

It's mainly responsible for three activities :-

- i) Loading
- ii) Linking
- iii) Initialization

i) Loading → The class loader reads the "class" file, generate the corresponding binary data & save it in the Method area. For each "class file", JVM stores the full information in the method area :-

- The fully additive name of the loaded class & its immediate parent class.
- Whether, "class" file is related to class or interface or Enum.
- Modifier, Variables & Method Information, etc.

After loading, JVM creates an object of type class to represent this file in the heap memory. To get this object, we can use `getClass()` method of object class.

ii) Linking → Performs verification, preparation & resolution.

- Verification ensure the correctness of the class file i.e. it checks whether this file is properly formulated & generated by a valid compiler or not.
- Preparation :- JVM collects memory for class static variables & initializing the memory to default values.
- Resolution :- It's the process of replacing symbolic references from the type with direct references.



iii) Initialization → All static variables are assigned with their values defined in the code & static block. This is executed in top to bottom order in a class and from parent to child in the class hierarchy.

There are 3 Class Loaders:-

- Bootstrap class loader
- Extension class loader
- System / Application class loader.

## PRACTICAL-02

AIM:- To implement following programs in Java

- i) Print your name
- ii) Perform Arithmetic operations
- iii) Convert Celsius to Fahrenheit & Fahrenheit to Celsius

as well as Information about variables & datatype in Java

## THEORY:-

Variable  $\rightarrow$  a variable is a symbolic name given to a value or data storage location. It allows you to store & manipulate data, making your code more flexible & dynamic

There are 3 types of variables:-

- i) Local variable  $\rightarrow$  A variable declared inside the body of the method is called local variable. A local variable can't be defined with "static"

Keyword

- ii) Instance variable  $\rightarrow$  A variable declared inside the class but outside the body of the method is called an instance variable. It's not declared as static.

- iii) static variable  $\rightarrow$  A variable that is declared as static is called static variable. It can't be local.



Datatype → A datatype in programming defines the type of data that a variable can hold. It specifies the kind of value that a variable can store.

### datatype

#### Primitive

It includes boolean, char, byte, etc.

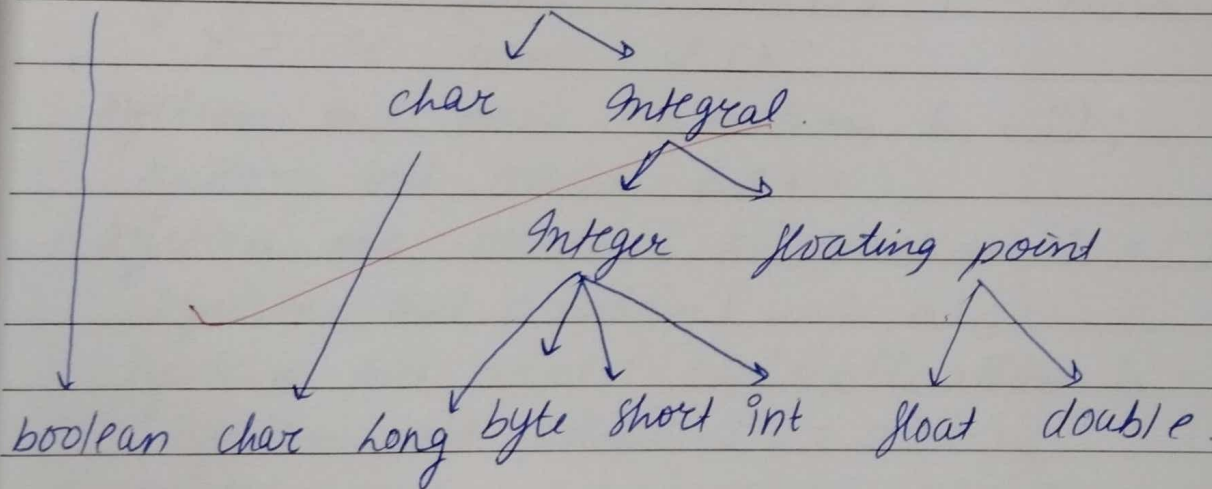
#### Non-primitive

It includes classes, Arrays, etc.

#### Boolean

#### Numeric

→ String  
→ Array, etc.



CODE :-

i) Print your name using Java

```

public class name {
    public static void main (String args[]) {
        System.out.println("shrishti-shrivastava");
    }
}
  
```

To compile : javac Name.java  
To execute :- java Name.

ii) To perform arithmetic operation in Java.

```
import java.util.Scanner;  
public class operator {  
    public static void main (String [] args) {  
        int a, b, c;  
        Scanner SC = new Scanner (System.in);  
        System.out.println ("Enter 1st No. :");  
        a = SC.nextInt ();  
        System.out.println ("Enter 2nd No. :");  
        b = SC.nextInt ();  
        System.out.print ("Addition is = ");  
        System.out.println (c = a + b);  
        System.out.print ("Subtraction is = ");  
        System.out.println (c = a - b);  
        System.out.print ("Multiplication is = ");  
        System.out.println (c = a * b);  
        System.out.print ("Division is = ");  
        System.out.println (c = a / b);  
        System.out.print ("Modulus is = ");  
        System.out.println (c = a % b);  
    }  
}
```



iii) To convert Fahrenheit to Celsius & Celsius to Fahrenheit.

```
import java.util.Scanner;
```

```
public class change {
```

```
    public static void main (String[] args) {
```

```
        float cel, Fah;
```

```
        Scanner Temp = new Scanner (System.in);
```

```
        System.out.println("Enter Temperature in Fahrenheit: ");
```

```
        Fah = Temp.nextFloat();
```

```
        System.out.println("Fahrenheit to Celsius: ");
```

```
        System.out.println(cel = (Fah - 32) * 5/9);
```

```
    }
```

```
}
```

CONCLUSION :- Hence, we study & implemented diff. types of programs also, we study about variables & datatypes.

~~Kathod~~  
24-08-23

\*\*\*\*\*

Date: / /  
\*\*\*\*\*

## PRACTICAL-03

AIM:- To implement the following programs.

- Java program to find whether the given NO. is even or odd.
- Java program to find a factorial of a NO.
- Java program to find fibonacci series.
- Java program to find average marks.

CODE :-

1] Write whether the NO. is given or odd

Java.

```
import util.Scanner;  
public class even {  
    public static void main (String args[]) {  
        Scanner SC = new Scanner (System.in);  
        int a;  
        System.out.println ("Enter the no.");  
        int a = SC.nextInt();  
        if (a%2 == 0) {  
            System.out.println ("The no. is even");  
        }  
        else {  
            System.out.println ("The no. is odd");  
        }  
    }  
}
```

\*\*\*\*\* ( LUCKY ) \*\*\*\*\*



## 2] Fibonacci Series :-

```

import java.util.Scanner;
public class fibonacci {
    public static void main (String args[]) {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter the no. of terms");
        int n = sc.nextInt();
        int n1 = 0, n2 = 1, n3, i, count = n;
        System.out.println ("The fibonacci series is");
        System.out.println (n1 + " " + n2);
        for (i = 2; i < count; i++) {
            n3 = n1 + n2;
            System.out.println (" " + n3);
            n1 = n2;
            n2 = n3;
        }
    }
}

```

## 3] Write a Program to find factorial of a No.

```

import java.util.Scanner;
public class factorial {
    public static void main (String args[]) {
        int i, fact = 1;
        System.out.println ("Enter the Number");
        Scanner sc = new Scanner (System.in);
    }
}

```

\*\*\*\*\*

)\*\*\*\*\*

```

int n = SC.nextInt();
for (i = 1; i <= n; i++) {
    fact = fact * i;
}

```

```

System.out.println("Factorial of the Number
is : " + fact);
}
}

```

4] Write a program to allocate grades for marks

```

import Java.util.Scanner;
public class grade {
    public static void main (String args []) {
        Scanner SC = new Scanner (System.in);
        int n;
        System.out.println ("Enter Marks");
        n = SC.nextInt();
        if (n >= 90) {
            System.out.println ("Grade A");
        }
        else if (n >= 70) {
            System.out.println ("Grade B");
        }
        else if (n >= 50) {
            System.out.println ("Grade C");
        }
    }
}

```

\*\*\*\*\* ( LUCKY ) \*\*\*\*\*



\*\*\*\*\*

\*\*\*\*\*

```
else {
```

```
    System.out.println("The student is fail");
```

```
}
```

```
}
```

```
}
```

CONCLUSION :- Hence we studied & implemented various programs in JAVA.

\*\*\*\*\* ( LUCKY ) \*\*\*\*\*