

JAVA

Java was developed by James Gosling at Sun Microsystems.

Java History is very interesting. it is a Programming Language. Created in 1991.

Sun microsystem Released its first Public implementation in 1996 as Java.

In 1997, Sun microsystem approached the ISO standards body and later formalized Java, but it soon withdrew from the process.

Java Hello World Program :-

A 'Hello world!' is a simple program that outputs Hello world! on the screen. Since it's a very simple program, it's often used to introduce a new programming language to a newbie.

```
// your first program
```

```
class HelloWorld {
    public static void main (String [] args) {
        System.out.println ("Hello, world!");
    }
}
```

Output - Hello world!

How Java 'Hello world!' Program works? !-

1. // your first Programme

In Java, any line starting with // is a comment. Comments are intended for users reading the code to understand the intent and functionality of the program.

2. class HelloWorld { }

In Java every application begins with a class

definition. In a program, HelloWorld is the name of the class, and the class definition is

```
class HelloWorld {  
  
}
```

for now, just remember that every java application has a class definition, and the name of the class should match the filename in java.

```
3- public static void main (String [] args) { ..... } -
```

This is the main method. Every application in java must contain the main method. The java compiler starts executing the code from the main method.

for now, just remember that the main function is the entry point of your java application, and it's mandatory in a java program. The signature of the main method in java is:

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


NATIONAL SKILL TRAINING INSTITUTE

Sion, Mumbai - 400 022.

Trade: _____ Unit No. _____ Page 2

Sub: _____ Lesson No. _____ Date _____

4. `System.out.println (" Hello world!");` -

The code above is a print statement. it prints the text Hello world! to standard output (your screen). The text inside the quotation marks is called string in java.

Notice the print statement is inside the main function, which is inside the class definition.

Things to take away! -

Every valid java application must have a class definition that matches the filename (class name and file name should be same).

The main method must be inside the class definition.

The compiler executes the codes starting from the main function.

THIS is the valid java program that does nothing.

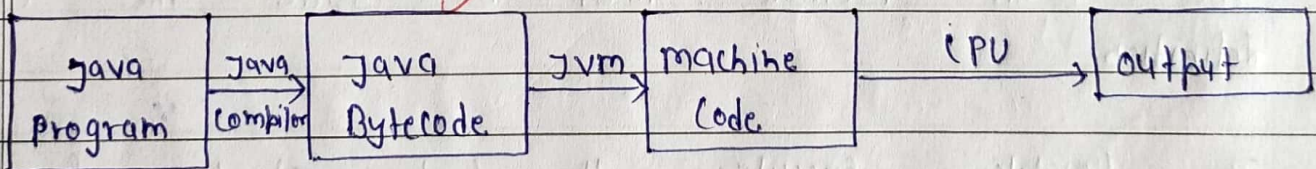
```
public class HelloWorld {  
  
    public static void main (String [] args) {  
  
        // write your code here  
  
    }  
  
}
```


JVM (Java Virtual Machine) :-

JVM (Java Virtual Machine) is an abstract machine that enables your computer to run a Java program.

When you run the Java program, the Java compiler first compiles your Java code to bytecode. Then the JVM translates bytecode into native machine code.

Java is a platform-independent language. It's because when you write Java code, it's ultimately written for JVM but not your physical machine (computer). Since JVM executes the Java bytecode which is platform-independent, Java is platform-independent.



JRE (Java Runtime Environment) :-

JRE (Java Runtime Environment) is a software package that provides Java class libraries, Java virtual machine (JVM), and other components that are required to run Java applications.

JRE is the superset of JVM.

$$\boxed{\text{JVM} + \text{Class Libraries} = \text{JRE}}$$

NATIONAL SKILL TRAINING INSTITUTE

Sion, Mumbai - 400 022.

Trade: _____ Unit No. _____ Page 3

Sub: _____ Lesson No. _____ Date _____

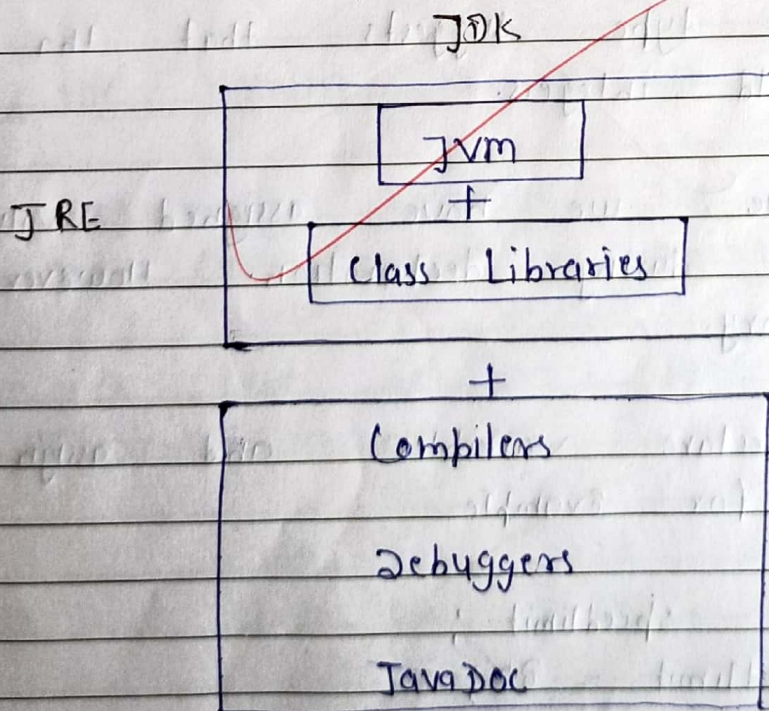
JDK (Java Development Kit) :-

JDK (Java Development Kit) is a software development kit required to develop applications in Java. When you download JDK, JRE is also downloaded with it.

In addition to JRE, JDK also contains a number of development tools (Compilers, JavaDoc, Java Debugger, etc.).

$$\boxed{\text{JDK} = \text{JRE} + \text{Compilers} + \text{Debuggers}}$$

Relationship between JVM, JRE and JDK :-



Excellent
[Signature]
10/10/22

Java Variables :-

A variable is a location in memory (storage area) to hold data.

To indicate the storage area, each variable should be given a unique name (identifier).

Create Variable in Java :-

```
int speedlimit = 80;
```

Here, speedlimit is a variable of int data type and we have assigned value 80 to it.

The int data type suggests that the variable can only hold integers.

In the example, we have assigned value to the variables during declaration. However, it's not mandatory.

You can declare variables and assign variables separately, for example

```
int speedlimit;  
speedlimit = 80;
```

Note - Java is a statically - typed language.

change values of variables :-

The value of a variable can be changed in the program, hence the name variable.

E.g.

```
int speedlimit = 80 ;
```

```
speedlimit = 90 ;
```

Here, initially, the value of speedlimit is 80. Later we changed it to 90.

```
int speedlimit = 80 ;
```

```
float speedlimit ;
```

Rules for Naming Variables in Java :-

Java Programming Language has its own set of Rules and Convention for naming variables.

- java is case sensitive. Hence 'age' and 'AGE' are two different variables.

E.g.

```
int age = 24 ;
```

```
int AGE = 25 ;
```

```
System.out.println (age) ; // Print 24
```

```
System.out.println (AGE) ; // Print 25
```


- Variables must start with either a letter or an underscore, `_` or a dollar, `$` sign.

E.g.

```
int age; // valid name and good practice
int_age; // valid but bad practice
int $age; // valid but bad practice
```

- Variable names cannot start with numbers.

Eg.

```
int 1age; // invalid variables
```

- Variable names can't use with space.

Eg.

```
int my age; // invalid variables
```

Here, if we need to use variables names having more than one word, use all lowercase letters for the first word and capitalize the first letter of each subsequent word, 'myAge'.

- when creating variables, choose a name that makes sense, for example score, number, level makes more sense than variables names such as s, n and l.

- if you choose one word variable names, use all lowercase letters, for example, it's better to use 'speed' rather than speed SPEED, or sPEED.

There are 4 Types of Variable in Java Programming Language -

- Instance variables (Non - static fields)
- Class variables (static fields)
- Local variables
- Parameters

Java literals :-

Literals are data used for representing fixed values. They can be used directly in the code.

E.g.

```
int a = 1;  
float b = 2.5;  
char c = 'F';
```

Here 1, 2.5 and 'F' are literals.

Here are different types of literals in java.

1- Boolean literals :-

In java Boolean literals are used to initialize Boolean datatypes. They can store two values: true and false.

E.g.

```
Boolean flag1 = false;  
Boolean flag2 = true;
```

Here, false and true are two Boolean literals.

2. Integer Literals ,

An integer literal is a numeric value (associated with numbers) without any fractional or exponential part. These are 4 types of integer literals in java -

- Binary (Base 2)
- Decimal (Base 10)
- Octal (Base 8)
- Hexadecimal (Base 16)

Eg.

```
// binary  
int binaryNumber = 0b10010;
```

```
// octal  
int OctalNumber = 027;
```

```
// decimal  
int decNumber = 34;
```

```
// Hexa decimal
```

```
int hexaNumber = 0x2F; // 0x Represent Hexadecimal
```

```
// binary
```

```
int binNumber = 0b10010; // 0b Represent binary
```

in java binary start with 0b, octal start with 0, and Hexa decimal start with 0x.

3. floating - Point Literals :-

A floating Points literal is a Numeric literal that has either a practical form or an exponential form.

e.g.

```
class main {
    public static void main ( string [] args) {
```

```
        double mydouble = 3.4 ;
```

```
        float myfloat = 3.4F ;
```

```
        // 3.44 * 10 ^ 2
```

```
        double mydoubleScientific = 3.445e2 ;
```

```
        system.out.println (mydouble) ; // Print 3.4
```

```
        system.out.println (myfloat) ; // Print 3.4
```

```
        system.out.println (mydoubleScientific) ; // Print 344.5
```

```
    }
```

```
}
```

output -

```
3.4
3.4
344.5
```

Note - The floating Point Literals are used to initialize 'float' and 'double' types variables.

4. Character literals :-

Character literals are Unicode character enclosed inside single quotes.

E.g.

```
char letter = 'a';
```

Here, 'a' is the character literal.

We can also use escape sequence as character literals.

For example, \b (backspace), \t (tab), \n (new line).

5. String literals :-

A string literal is a sequence of character enclosed inside double - quotes.

E.g.

```
String str1 = "Java Programming";  
String str2 = "Programize";
```

Here, Java Programming and Programize are two string literals.

Java data types (Primitive) :-

As the name suggests, data type specify the type of data that can be stored inside variable in Java.

Java is a statically - typed language. This means that all variables must be declared before they can be used.

```
int speed;
```

Here 'speed' is a variable, and the data type of the variable is int.

The int datatype determine that the speed variable can only contain integers.

There are 8 data types predefined in Java, know as Primitive data types.

1. Boolean data type :-

The Boolean data type has two possible values, either true or false.

Default value : false.

They are usually used for true/false conditions.

E.g.

```
class main {  
    public static void main (String [] args) {  
        Boolean flag = true;  
        System.out.println (flag); // prints true  
    }  
}
```



```
}  
}
```

O/P - true

2. Byte data type :-

The byte data type can have values from -128 to 127 (8-bit signed two's complement integer).

if it's certain that the value of a variable will be within -128 to 127, then it is used instead of int to save memory.

Default value = 0

e.g.

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

```
        byte range ;  
        range = 124 ;
```

```
        System.out.println (range) ; // print 124  
    }  
}
```

output - 124

3. Short data type :-

The short data type is in java can have values from -32768 to 32767 (16-bit signed two's complement integer).

if it's certain that the value of a variable will be within -32768 and 32767, then it is used instead of other integer data types (int, long).

default value = 0

Eg.

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

```
        short temperature ;
        temperature = -200;
        System.out.println (temperature) ; // Print -200
    }
}
```

output - -200

4. int data type :-

The int data type can have values from -2^{31} to $2^{31}-1$ (32-bit signed two's complement integer)

if you are using java 8 or later, you can use an unsigned 32-bit integer. This will have a minimum value of 0 and a maximum value of $2^{32}-1$.

default value - 0

E.g.

```
class main {  
    public static void main (String [] args) {  
  
        int range = -4250000 ;  
        System.out.println (range) ; // Print - 4250000  
    }  
}
```

O/P - -4250000

5. Long data type :-

The long data type can have values from -2^{63} to $2^{63}-1$ (64-bit signed two's complement integer).

default value - 0

E.g.

```
class LongExample {  
    public static void main (String [] args) {  
  
        long range = -42332200000L ;  
        System.out.println (range) ;  
    }  
}
```

O/P - -42332200000L

6. Double data type :-

The double data type is a double precision 64-bit floating - point.

it should never be used for precise values such as currency.

default value - 0.0 (0.0d)

E.g.

```

class main {
    public static void main (String[] args) {
        double number = -42.3;
        System.out.println (number); // Print -42.3
    }
}
    
```

O/P - -42.3

Good
Muhammad
10/06/22

7. float data type :-

The float data type is a single - precision 32-bit floating - point.

It should never be used for precise values such as currency.

default value - 0.0 (0.0f)

E.g.

```
class main {  
    public static void main (String [] args) {  
        float number = -42.3 f ;  
        System.out.println (number) ; // Print -42.3  
    }  
}
```

O/P - -42.3

Q- char data type :-

it's a 16-bit Unicode character.

E.g.

```
class main {  
    public static void main (String [] args) {  
        char letter 1 = '9' ;  
        System.out.println (letter 1) ; // print 9  
        char letter 2 = 65 ;  
        System.out.println (letter 2) ; // print A  
    }  
}
```

O/P - 9A

Here, we have assigned 9 as a character (single quote) to the letter 1 variable. However, the letter 2 variable is assigned 65 as a integer No. (No single quote).

String data type :-

Java also provides support for character string via `java.lang.String` class. Strings in Java are not primitive types. Instead, they are objects.

Eg.

```
String myString = "Java Programming";
```

Here, `myString` is an object of the `String` class.

JAVA Operators :-

Operators are symbols that perform operations on variables and values. For example, `+` is an operator used for addition, while `*` is also an operator used for multiplication.

Operator in Java can be classified into 5 types.

1. Arithmetic Operators
2. Assignment Operators
3. Relational Operators
4. Logical Operators
5. Bitwise Operators

1. Java Arithmetic Operators :- Arithmetic operators are used to perform arithmetic operations on variables and data.