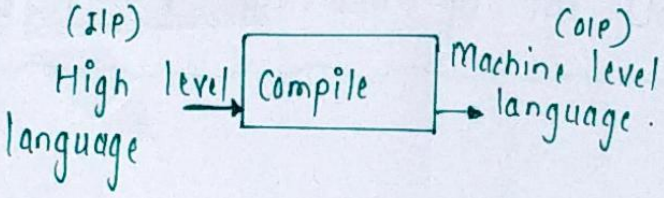
What is Programming ?

- Programming is an art of having a Conversation with the Computer.
- Basically Computers does not understand Hindi, Marathi, Kannada, English Or any High level Language then which language Computers understand ?
- Computers understands Only Machine level language or Binary language i.e 0 or 1.

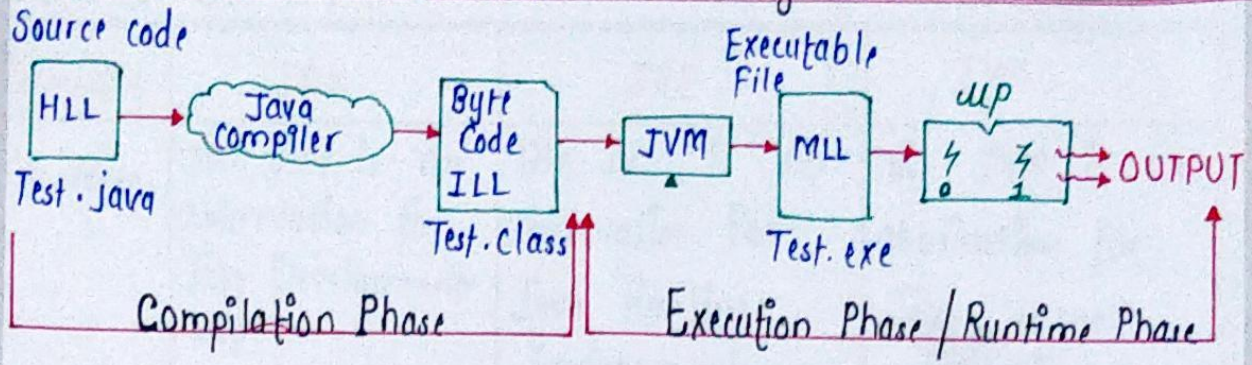
0 ⟶ Absence of charge
1 ⟶ presence of charge

- A Software is required to convert this High level language into the Machine level language.

Compiler is a software which is used to convert High level language into the machiene level language.

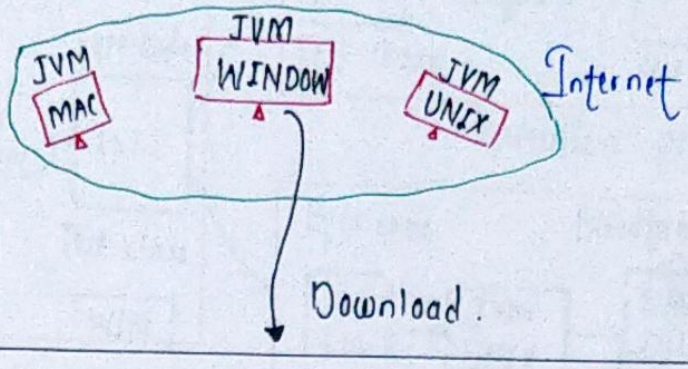


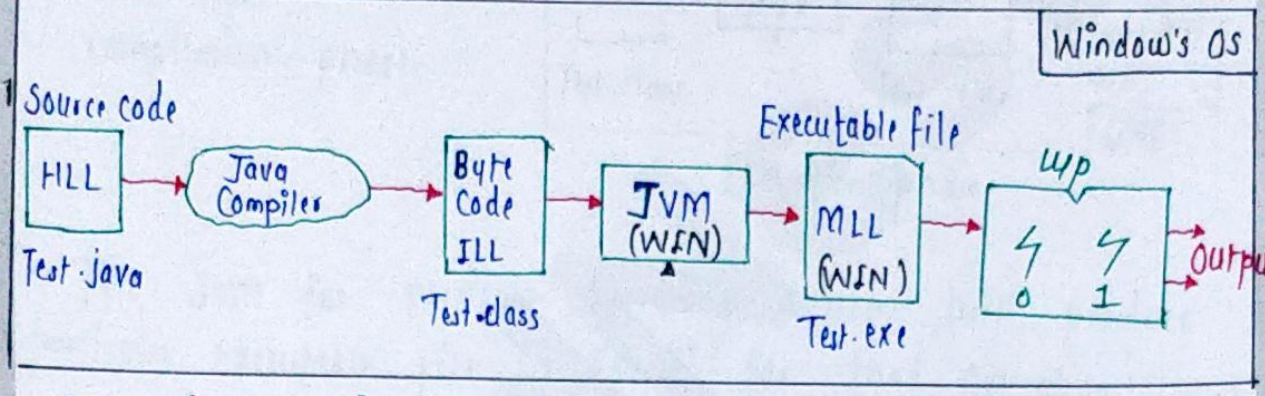
* Flow of Execution of Java Program :-



- Hight level language is understanable by Humans & this is Source code file i.e Test.java is in HLL.
- Java compiler is used to Convert this Test.java file into Machiene level language, but java compiler convert this file into ILL (intermediate level language) which is neither in HLL nor in MLL.
- To get Executable file JVM converts Byte code (ILL file) into Machiene level language (Executable file, Test.exe).
- Executable file is given to the CPU, processing will happen & finally we will get the Output.

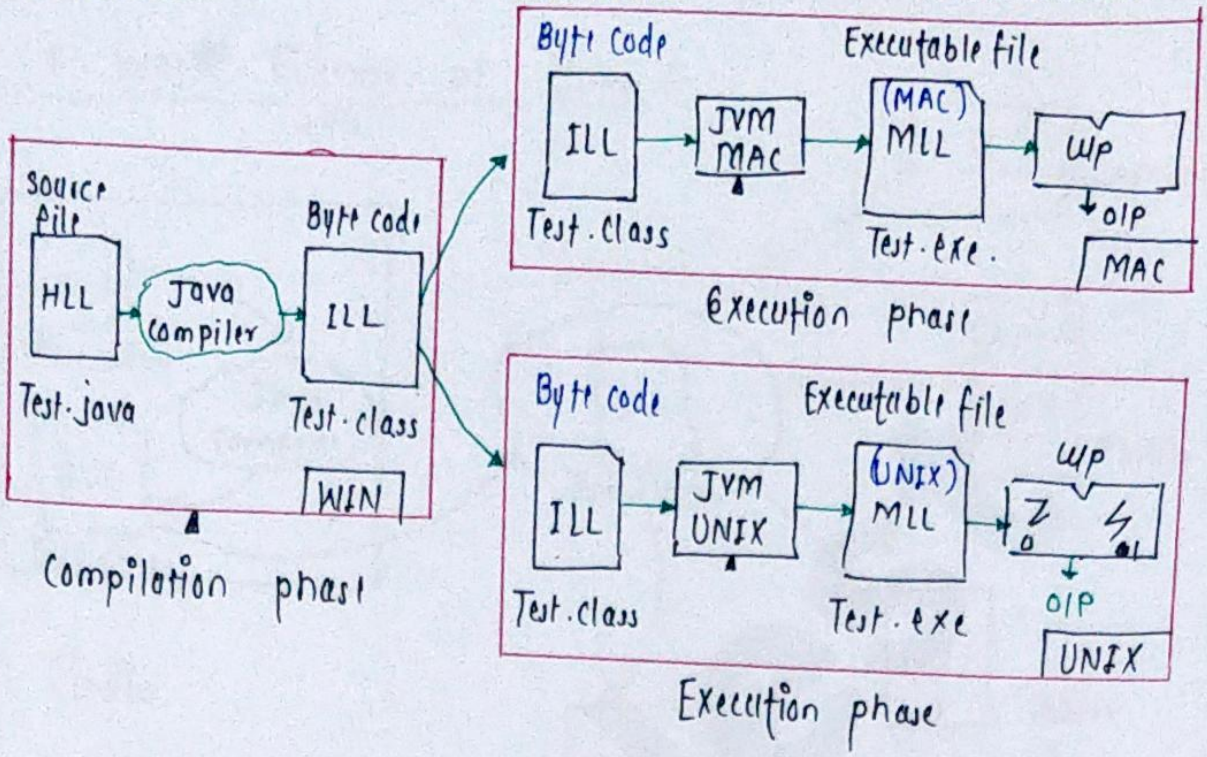
* WORA feature of Java :-





- JVM is specific for windows Operating system & will be producing an executable file specific for windows
- up will process the executable file & gives the o/p.

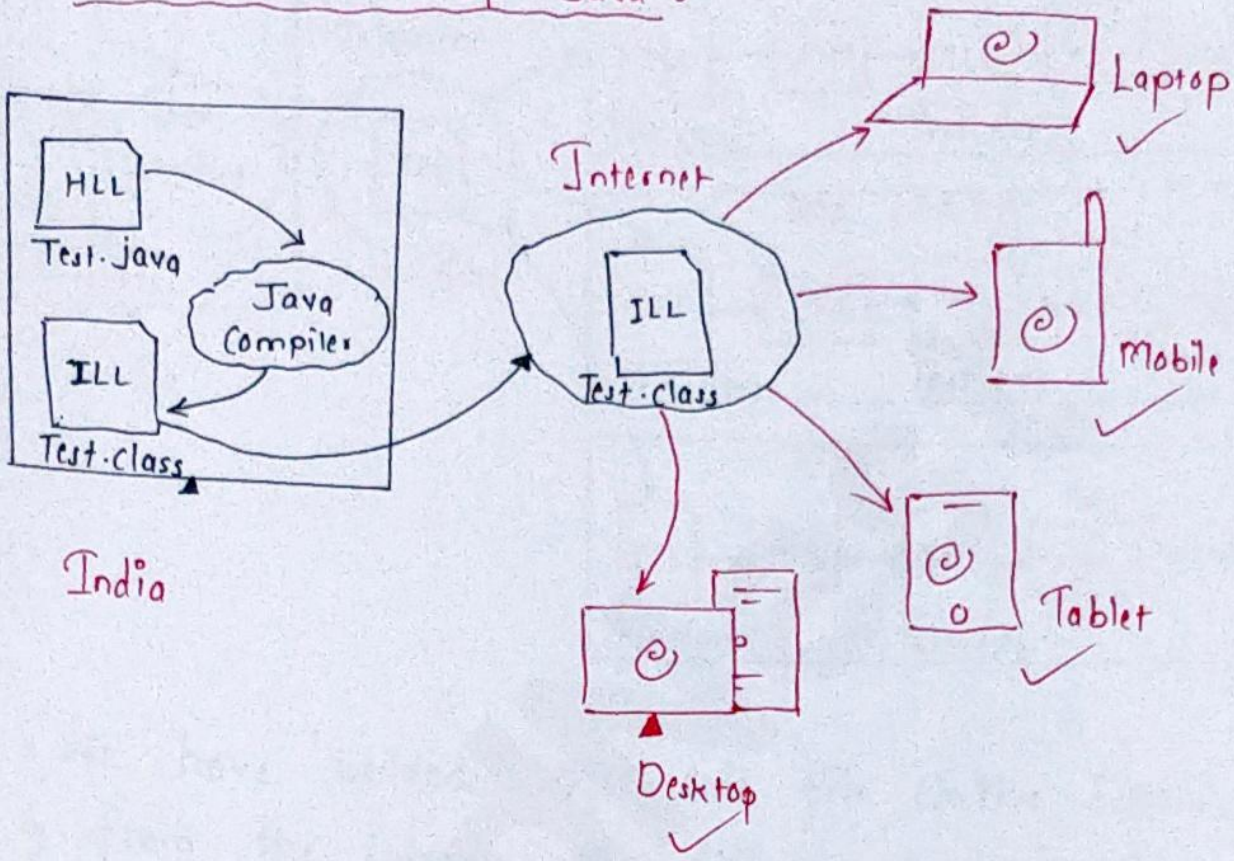
* A programming Language is Considered as portable if it is Capable of Compiling and executing on different platforms.
- For e.g Consider three different Computers having different Operating systems such as Window's, MAC, UNIX respectively.



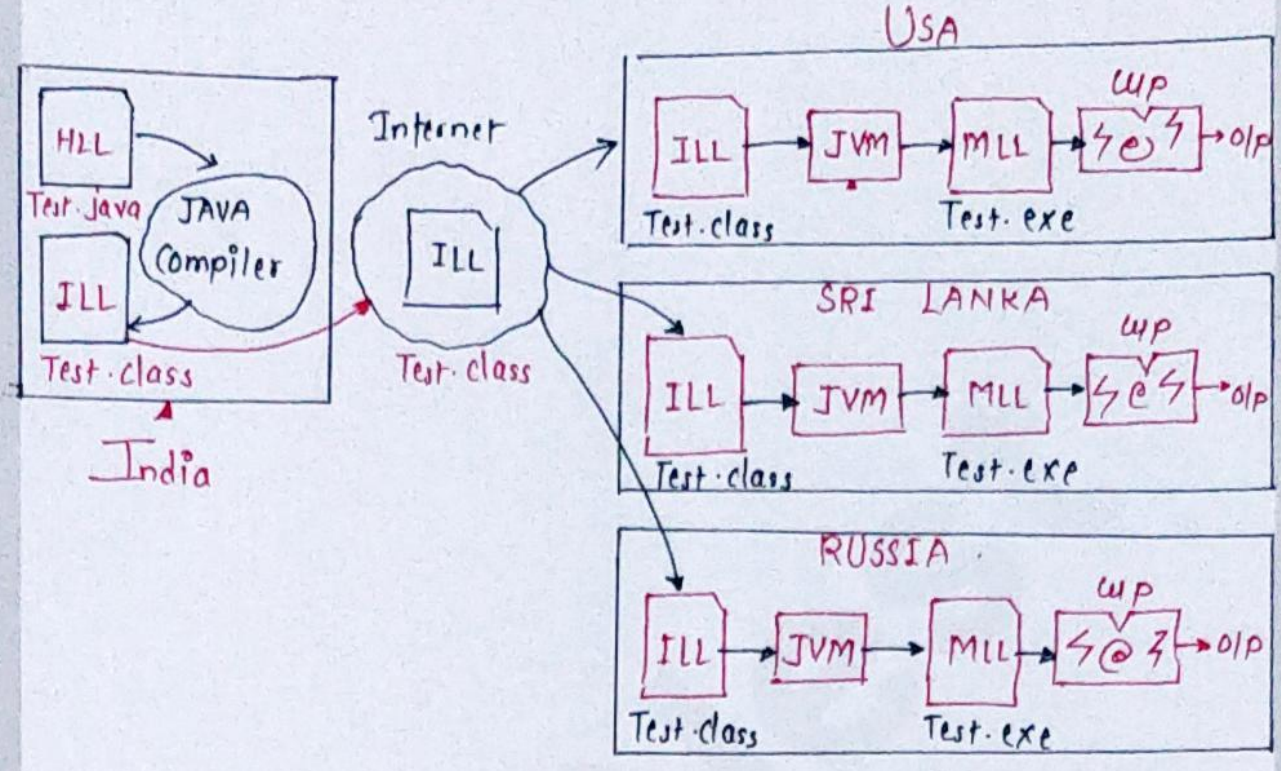
- The JVM for particular Operating system will produce particular executable file i.e specific for that Operating system.
- JVM for MAC will only be producing an executable file capable of executing on MAC & same JVM for UNIX will only be producing an executable file capable for executing on UNIX.
- Finally wp will give the o/p.

* We are able to compile java program on one platform & we are also able to execute it on several other platforms.
* This is the reason why java is considered as portable programming Language or platform independent Language.

* WORA Feature of Java :-



- JAVA is Architecture Neutral. i.e we can compile java code on one type of Device & we can execute it on diffrent types of Devices.
- Byte code file if we uploaded on the internet different devices will be able to execute the java program.



- We have uploaded Byte code file On the internet & from the internet the different computers from the different countries will try to download that byte code file.
- JAVA program Can be Compiled at One geographical location and Can be executed On different geographical locations. as well.
- We have written the code, Compiled the code Once & reading it multiple times.

Introduction to Programming

What is Programming?

Instructing the processor in order to tell what has to be done. For example addition, subtraction, multiplication, division etc.

Programming is an art.

If you want to cook, you need raw material. for Cooking.

If you want to write a Program then you also need raw materials for programming.

Raw-material of Programming

Syntax of Programming

```
for (Initialization; Condition; Iteration)
    // int a=0, b=0   a<5   a++, b--
{
    // Body // statements
}
```

Conditional Control Constructs

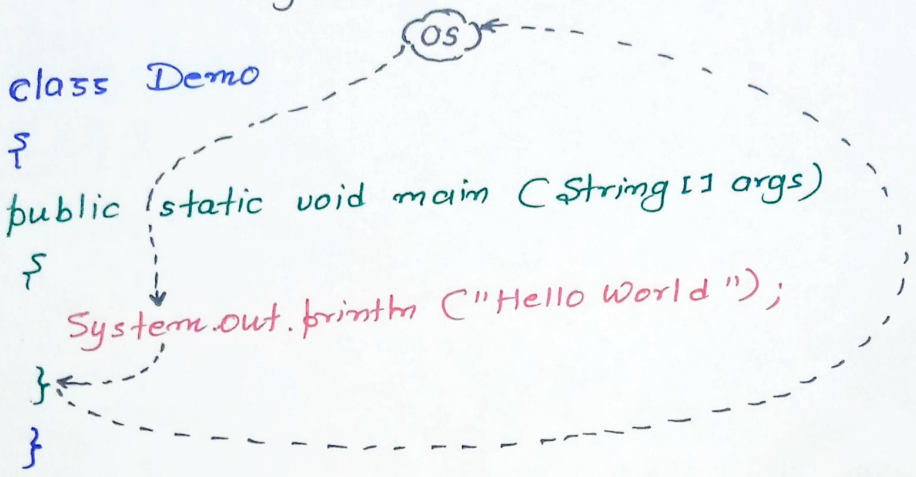
Simple if
else if
else if-ladder
switch

Looping Control Constructs

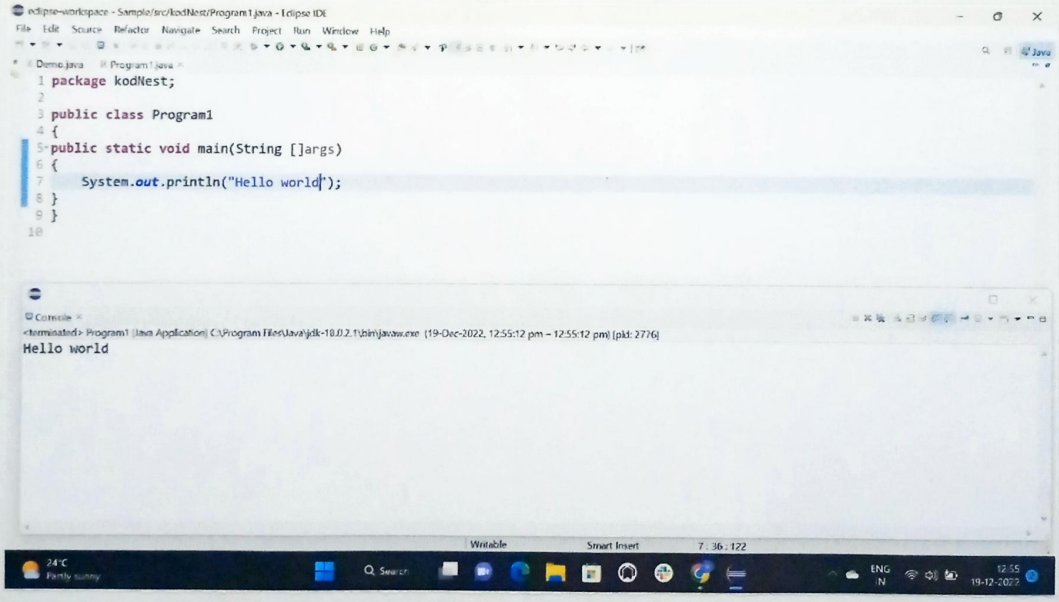
for
while
do-while
enhanced for loop
labbed for loop

Break return go-to

Write a Java Program to print Hello world on to monitor ?



OUTPUT

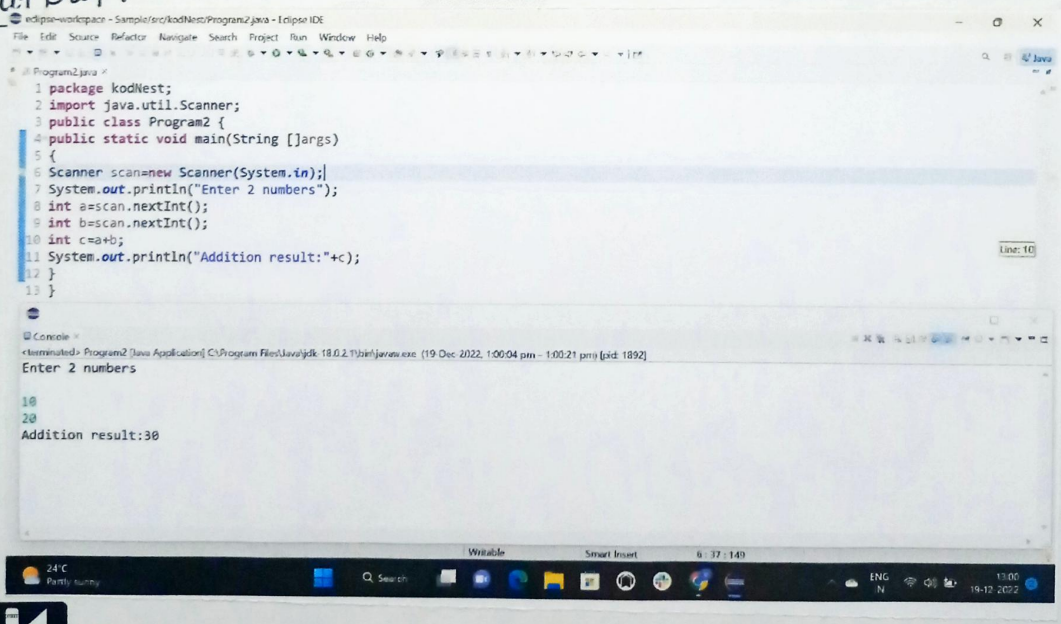


Write a Java program for addition of 2 numbers by taking input from the keyboard ?

```
import java.util.Scanner;
class Demo
{
public static void main (String [] args)
{
Scanner scan = new Scanner (System.in);
System.out.println ("Enter 2 numbers");
int a = scan.nextInt ();
int b = scan.nextInt ();
int c = a+b;
System.out.println ("Addition result:" + c);
}
}
```

OS

Output:

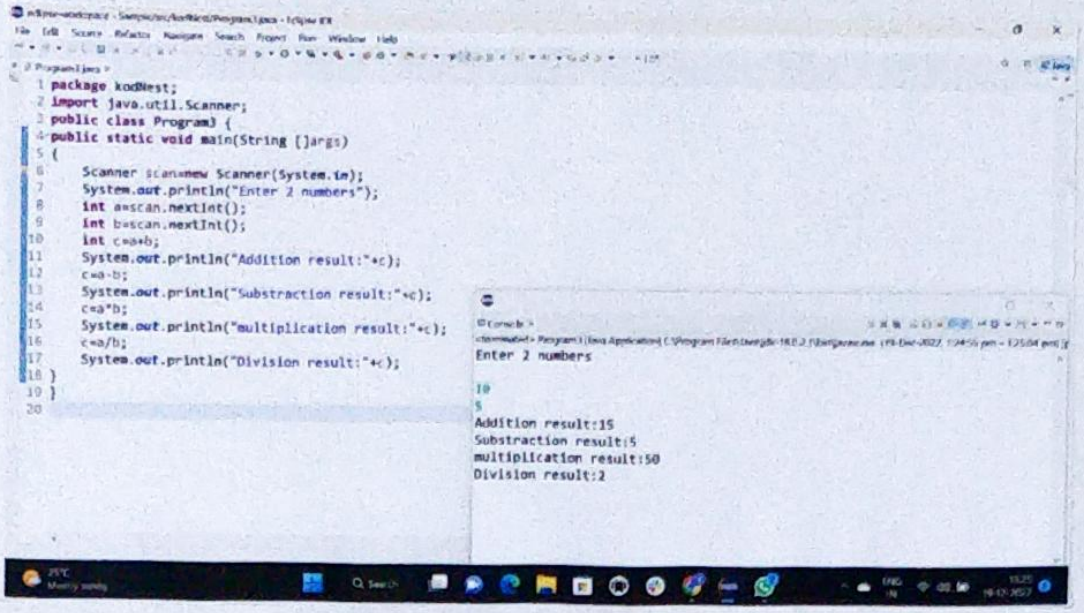


Write a Java program to perform Addition, substraction, multiplication and Division on 2 numbers by taking the input from the keyboard ?

```
import java.util.Scanner;
public class Demo {
public static void main (String[] args)
{
Scanner scan = new Scanner (System.in);
System.out.println ("Enter 2 numbers");
int a = scan.nextInt();
int b = scan.nextInt();
int c = a+b;
System.out.println("Addition result:" +c);
c= a-b;
System.out.println(" Substraction result:" +c);
c = a*b;
System.out.println("Multiplication result:" + c);
c = a/b;
System.out.println(" Division result:" +c);
}
}
```

OS

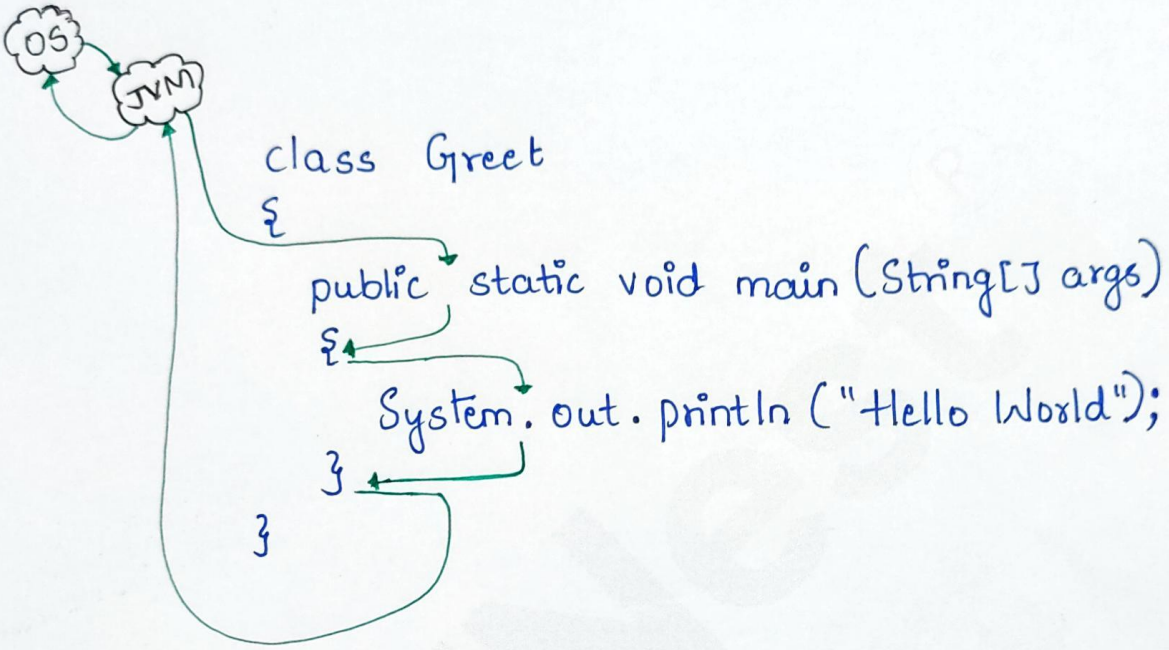
Output



```
package kodNest;
import java.util.Scanner;
public class Program1 {
public static void main(String []args)
{
    Scanner scan=new Scanner(System.in);
    System.out.println("Enter 2 numbers");
    int a=scan.nextInt();
    int b=scan.nextInt();
    int c=a+b;
    System.out.println("Addition result:"+c);
    c=a-b;
    System.out.println("Substraction result:"+c);
    c=a*b;
    System.out.println("multiplication result:"+c);
    c=a/b;
    System.out.println("Division result:"+c);
}
}
```

```
Enter 2 numbers
10
5
Addition result:15
Substraction result:5
multiplication result:50
Division result:2
```

Java Program To Print "Hello world"



output :-

```
Hello World
```

Note :-

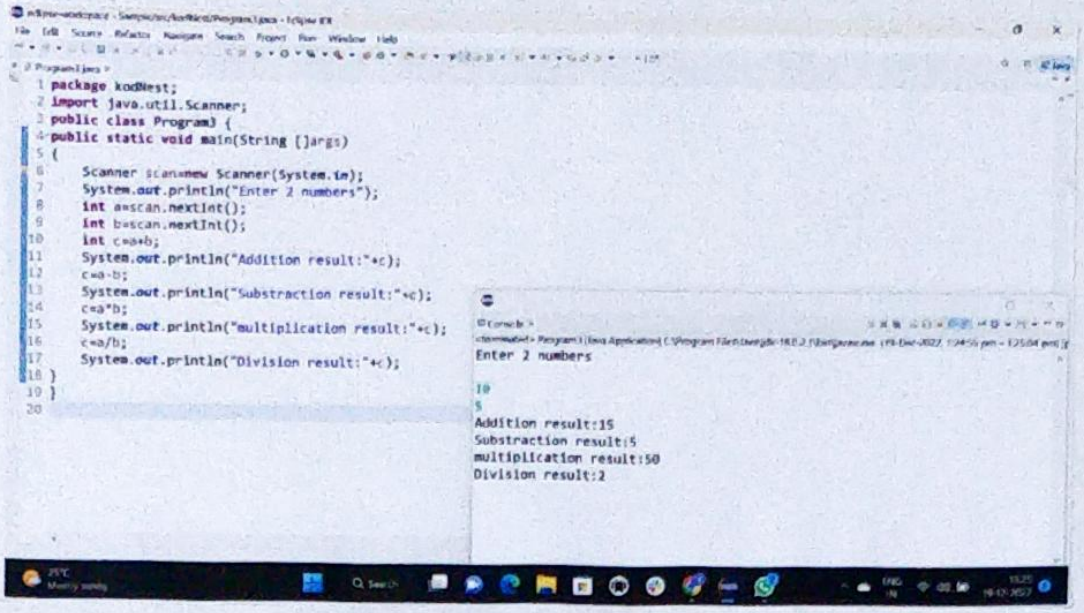
Always save Java files using .java extension.

Write a Java program to perform Addition, substraction, multiplication and Division on 2 numbers by taking the input from the keyboard ?

```
import java.util.Scanner;
public class Demo {
public static void main (String[] args)
{
Scanner scan = new Scanner (System.in);
System.out.println ("Enter 2 numbers");
int a = scan.nextInt();
int b = scan.nextInt();
int c = a+b;
System.out.println("Addition result:" +c);
c= a-b;
System.out.println(" Substraction result:" +c);
c = a*b;
System.out.println("Multiplication result:" + c);
c = a/b;
System.out.println(" Division result:" +c);
}
}
```

OS

Output



```
package kodNest;
import java.util.Scanner;
public class Program1 {
public static void main(String []args)
{
    Scanner scan=new Scanner(System.in);
    System.out.println("Enter 2 numbers");
    int a=scan.nextInt();
    int b=scan.nextInt();
    int c=a+b;
    System.out.println("Addition result:"+c);
    c=a-b;
    System.out.println("Substraction result:"+c);
    c=a*b;
    System.out.println("multiplication result:"+c);
    c=a/b;
    System.out.println("Division result:"+c);
}
}
```

```
Enter 2 numbers
10
5
Addition result:15
Substraction result:5
multiplication result:50
Division result:2
```

Conditional Control Construct

Simple if

```
import java.util.Scanner;
class Demo
{
public static void main (String [] args)
{
Scanner scan = new Scanner (System.in);
System.out.println (" Press the key-1 on the keyboard ");
int n = scan.nextInt ();
if (n == 1)
{
System.out.println (" You have pressed 1 on the keyboard ");
}
System.out.println (" You have not pressed 1 on the keyboard ");
}
}
```

Output :

```
package kodNest;
import java.util.Scanner;
public class Program4 {
public static void main(String []args)
{
    Scanner scan=new Scanner(System.in);
    System.out.println("Press the key-1 on the keyboard");
    int n=scan.nextInt();
    if(n==1)
    {
        System.out.println("You have Pressed 1 on the Keyboard");
    }
        System.out.println("You have not Pressed 1 on the Keyboard");

}
}
```

```
Press the key-1 on the keyboard

1
You have Pressed 1 on the Keyboard
You have not Pressed 1 on the Keyboard
```

Data Types in Java

Real World Data	Java Data Types
Integers Ex :- (20, 25, 35 etc...,)	byte, short, int, long
Real / Fractional Numbers Ex :- (5.5, 5.7...)	float, double
Characters Ex :- (d, c, c, p...)	char
Yes - No Ex :- (Married - Unmarried)	boolean
Multimedia Ex :- (Audio, Video, Photo etc...)	Non-Primitive data types

Note :- byte, short, int, long, float, double, char and boolean are primitive datatypes

DataType to deal with Integer

Data Type	Size	Range
byte	1 byte	-128 to 127
short	2 bytes	-32,768 to 32,767
int	4 bytes	-2,147,483,648 to 2,147,483,647
long	8 bytes	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807

Example :-

```
Class DataTypesDemo
{
    public static void main(String args[])
    {
        byte age = 35;
        System.out.println("Age = " +age);

        Short year =2023;
        System.out.println("Year = " + year);
```

```
        int sal = 365000;
        System.out.println("Salary = " + sal);

        long pop = 999999999991;
        System.out.println("Population = " + pop);
    }
}
```

Output :-

```
Age = 35
Year = 2023
Salary = 365000
Population = 999999999999
```

Datatypes to deal with Real Numbers.

To deal with real numbers or fractional numbers or decimal numbers java provides 2 data types

1. float data type
2. double data type.

float data type :

→ float data type occupies 4 bytes of memory.

→ float data type can be used when we want the precision after the decimal point more than 5-7 digits.

Program using float

```
class DataTypesDemo
{
    public static void main (String [] args)
    {
        float gravity = 9.8f;
        System.out.println ("Gravity = "+gravity);
    }
}
```

Output:

Gravity = 9.8

Note: While storing a number inside float type variable we must use the abbreviation 'f' or 'F'.

double datatype.

→ double data type occupies 8 bytes of memory.

→ double datatype can be used when we want the precision after the decimal point more than 14 - 16 digits.

Program on double data type.

```
class Demo
{
    public static void main (String [] args)
    {
        double pi = 3.141592653589;
        System.out.println("pi = "+pi);
    }
}
```

Output

pi = 3.141592653589.

Char Data Type

To store character type data in java we have a data type called as "char".

"char" data types occupies 2 bytes in memory.

> Note: if we want to store multiple characters such as strings then, java provides us with something called as "String" class.

To store single character we have "char" data type.

Program:

```
class DataTypesDemo
{
    Public static void main (String [] args)
    {
        char gen = 'M';
        System.out.println ("Grender : "+gen);
    }
}
```

Output:

```
Grender : M
```

Datatype to deal with True-False.

To store true or false data, java provides us "boolean" data type.

Memory occupied by boolean is JVM dependent

Program.

```
class DataTypesDemo
{
    public static void main(String[] args)
    {
        boolean isMarried = false;
        System.out.println("Are you Married?"
                           + isMarried);
    }
}
```

Output:

Are you married? false.

Type Casting.

Type casting is a process of storing the value of one datatype into another datatype

There are two types of type casting:

1. Implicit type casting
2. Explicit type casting.

1. Implicit Typecasting.

Converting or storing a value of smaller data type into the variable of bigger data type is called as Implicit typecasting.

Program:

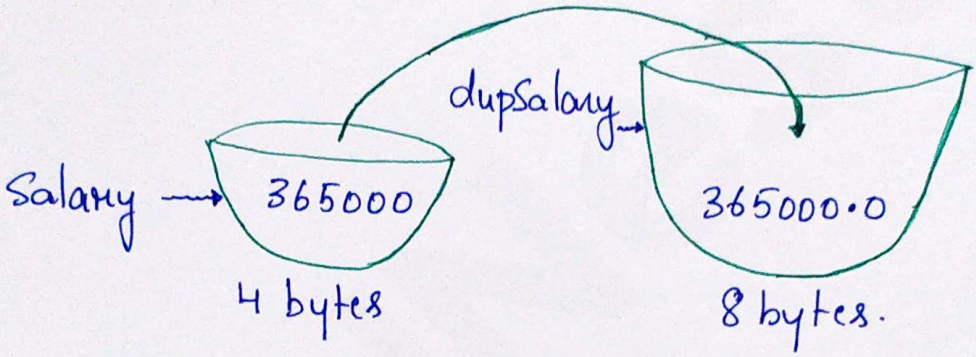
```
class TypeCastingDemo
{
   public static void main(String[] args)
   {
        int salary = 365000;
        double dupSalary = salary;

        System.out.println("Salary = "+salary);
        System.out.println("dupSalary = "+dupSalary);
   }
}
```

Output:

```
Salary    = 365000
dupSalary = 365000.0
```

Memory Map:

```
int Salary = 365000
double dupSalary = Salary.
```



Note: Here we are storing the value of int type variable inside the double type variable. Hence the value is getting stored into double type variable, extra .0 will be added as a fractional part.

⑤ Explicit Type Casting:

Converting or storing a value of bigger data type into the variable of smaller data type. is called as Explicit Type Casting.

Program:

```
class TypeCastingDemo
{
    public static void main(String[] args)
    {
        double pi = 3.14159;
        int dupPi = pi;  ~~~> Error.

        System.out.println("Pi = "+pi);
        System.out.println("Duplicate pi = "+dupPi);
    }
}
```

Note: Here we are trying to store the value of double inside the variable of int directly without performing Explicit type casting.

Program using Explicit type Casting.

```
class TypeCastingDemo
{
    public static void main(String[] args)
    {
        double pi = 3.14159;
        int dupPi = (int)pi;

        System.out.println("Pi = "+pi);
        System.out.println("Duplicate pi ="+dupPi);
    }
}
```

Output:

```
Pi = 3.14159
Duplicate pi = 3.
```

Note: Hence int datatype doesn't store decimal values, fractional part is not stored in int variable while Explicit conversion.

if-else

```
import java.util.Scanner;

class Demo
{
public static void main (String[] args)
{
Scanner scan = new Scanner (System.in);
System.out.println ("Press the key-1 on the keyboard");
int n = scan.nextInt ();
if (n == 1)
{
System.out.println ("You have pressed 1 on the keyboard");
}
else
{
System.out.println ("You have not pressed 1 on the keyboard");
}
}
}
```

OS

T

Output :

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

        Scanner scan = new Scanner(System.in);
        System.out.println("Press The Key-1 on the keyboard");
        int n=scan.nextInt();

        if(n==1)
```

```
Press The Key-1 on the keyboard
1
you have pressed 1 on the keyboard
```

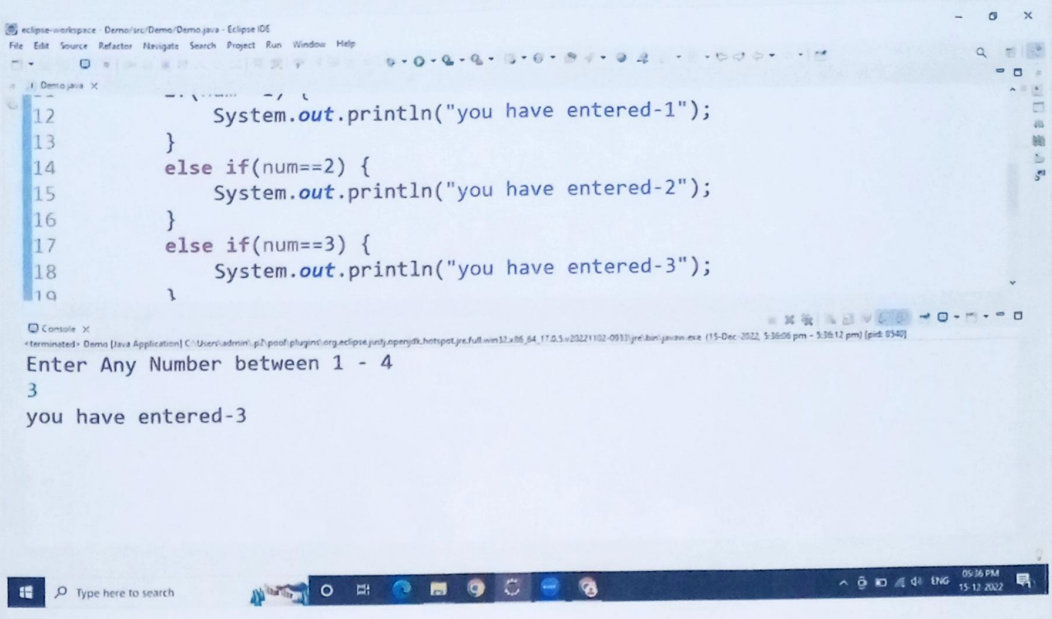
```
public class Demo {
    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        System.out.println("Press The Key-1 on the keyboard");
        int n=scan.nextInt();

        if(n==1)
```

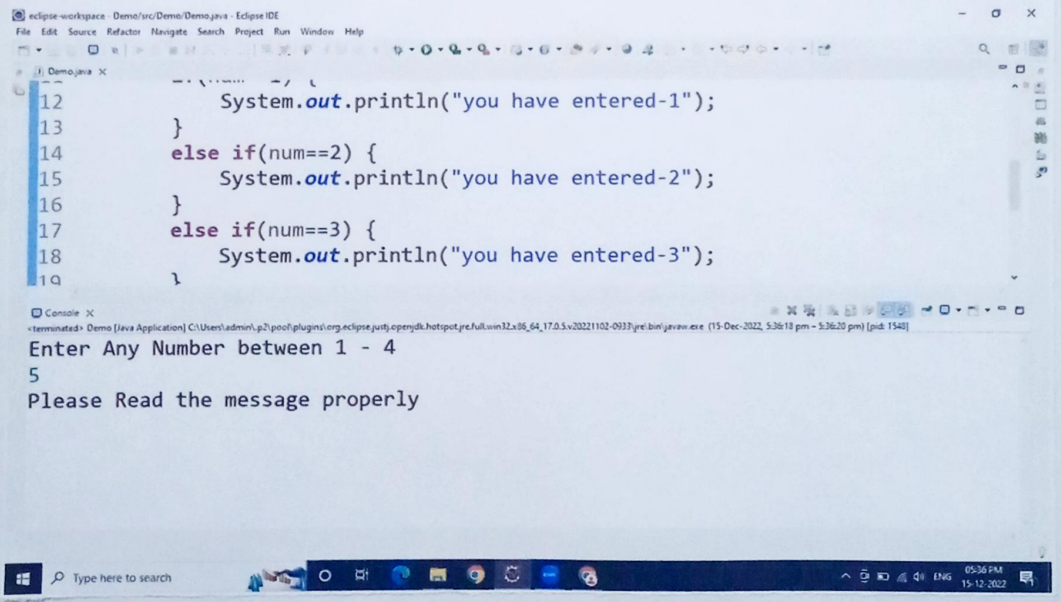
```
Press The Key-1 on the keyboard
2
you haven't pressed 1 on the keyboard
```

else if ladder

```
class Demo
{
public static void main(String[] args)
{
Scanner scan = new Scanner(System.in);
System.out.println("Enter any number b/w 1-4");
int num = scan.nextInt();
if(num==1)
{
System.out.println("You have entered 1");
}
else if(num==2)
{
System.out.println("You have entered 2");
}
else if(num==3);
{
System.out.println("You have entered 3");
}
else if(num==4)
{
System.out.println("You have entered 4");
}
else
{
System.out.println("Please read the msg properly");
}
}
}
```

Output :


eclipse-workspace - Demo/src/Demo/Demo.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Demo.java
12 System.out.println("you have entered-1");
13 }
14 else if(num==2) {
15 System.out.println("you have entered-2");
16 }
17 else if(num==3) {
18 System.out.println("you have entered-3");
19 }
Console
<terminated> Demo [Java Application] C:\Users\admin\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (15-Dec-2022, 5:36:06 pm – 5:36:12 pm) [pid: 8540]
Enter Any Number between 1 - 4
3
you have entered-3
Type here to search
ENG 05:36 PM 15-12-2022


eclipse-workspace - Demo/src/Demo/Demo.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Demo.java
12 System.out.println("you have entered-1");
13 }
14 else if(num==2) {
15 System.out.println("you have entered-2");
16 }
17 else if(num==3) {
18 System.out.println("you have entered-3");
19 }
Console
<terminated> Demo [Java Application] C:\Users\admin\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (15-Dec-2022, 5:36:18 pm – 5:36:20 pm) [pid: 1540]
Enter Any Number between 1 - 4
5
Please Read the message properly
Type here to search
ENG 05:36 PM 15-12-2022

Java Program To Calculate The Average Of 3 Integers

```
class AverageCalculator
{
    public static void main(String args[])
    {
        int n1=10;
        int n2=20;
        int n3=50;

        int sum=n1+n2+n3;
        double avg=sum/3.0;
        System.out.println("Average "+avg);
    }
}
```

Output

```
Average 26.666666666666668
```

Data Type

Being a programmer we never should hard code our programs instead of we take the input from user by making use of Scanner class.

When ever we make use of Scanner class in any programs we must import it from a package called java.util.

Example

```
import java.util.Scanner;
class AverageCalculator {
    public static void main (String args[])
    {
        Scanner scan = new Scanner (System.in);
        int n1 = scan.nextInt();
        int n2 = scan.nextInt();
        int n3 = scan.nextInt();

        int sum = n1+n2+n3;
        double avg = sum/3.0;
        System.out.println ("Average " + avg );
    }
}
```

Output :

```
20
50
80
Average 50.0
```

Example For User Input With The Help Of Scanner

```
import java.util.Scanner;
  class UserInputDemo
  {
    public static void main(String args[])
    {
      Scanner scan = new Scanner(System.in);
      System.out.println("Please enter age");
      byte age= scan.nextByte();
      System.out.println("Age = "+ age);

      System.out.println("Please enter your height");
      float height = scan.nextFloat();
      System.out.println("Height =" + height);
    }
}
```

Output

```
Please enter age
35
Age = 35

Please enter your height
7.5
Height = 7.5
```

Introduction To Object-Oriented Programming Language

Object-oriented programming language is used to solve real-world problems.

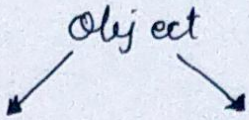
Objects:- Are real-world entities. That is anything which is present in the real world is considered an object.

All the instructions which are required to create an object in java must be kept inside a class.

Object Orientation Programming

example-1

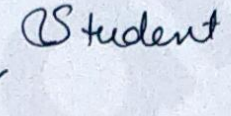
- To create an object we require the following information.



State of object
or
Property of object

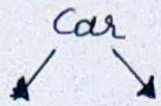
Behaviour of Object
or
Actions of Objects

For Example:-



state of student
- name
- roll no
- marks
- skills

behaviour of Student.
- studying
- playing
- sleeping
- eating.



- brand
- color
- price

- accelerates

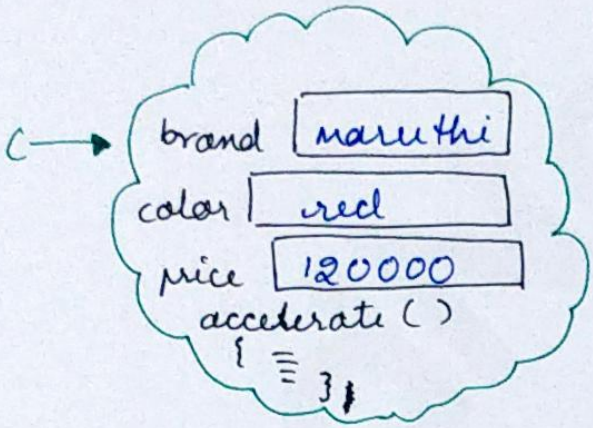
```
class Car
{
    String brand;
    String color;
    int price;

    void accelerate()
    {
        System.out.println("A car accerates");
    }
}
class CarApp
{
public static void main(String []args)
{
    Car c = new Car();
    c.brand = "maruthi";
    c.color = "red";
    c.price = 120000;
```

state / properties.

behaviour / action performed.

```
        System.out.println("Brand: "+c.brand);
        System.out.println("Color: "+c.color);
        System.out.println("Price: "+c.price);
        c.accelerate();
    }
}
```

In heap memory :-



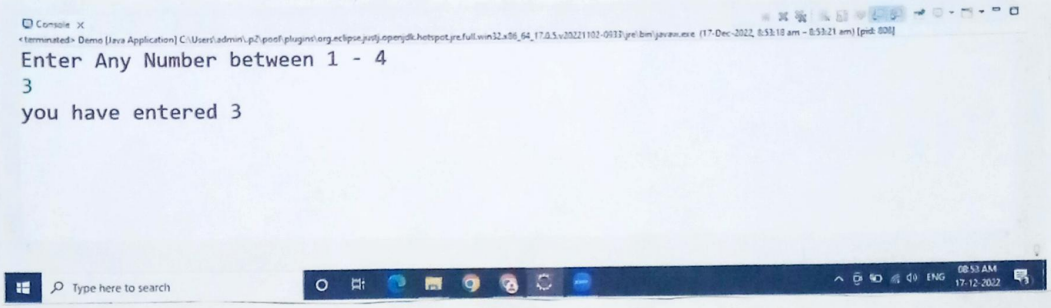
Output :-

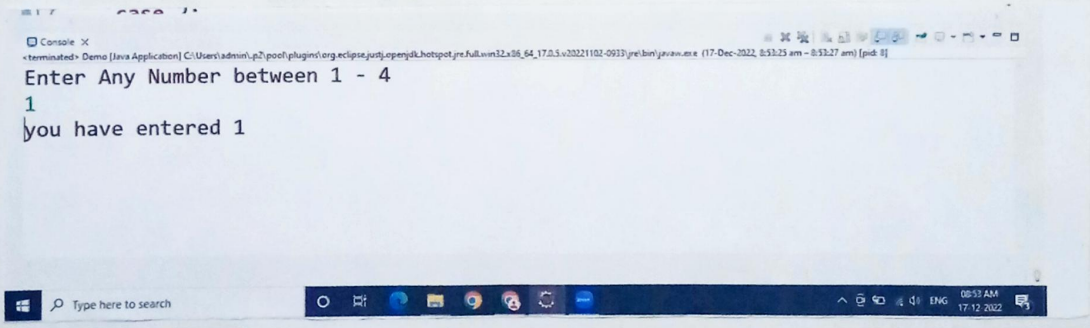
for compilation :- javac CarApp.java
for execution :- java CarApp

```
Brand : maruthi
Color : red
Price : 120000.
A car accelerates.
```

```
Import java.util.Scanner;
Class Demo{
Public Static void main(String []args){
Scanner scan=new Scanner(System.In);
System.out.println("Enter Any Number b/w
                    1-4");
        Int num = Scan.nextInt();
  Switch(num)
   {
    case1:
System.out.println("you have entered-1");
     break;
   case 2:
System.out.println("you have entered-2");
     break;
    case 3:
System.out.println("you have entered-3");
     break;
    case4:
System.out.println("you have entered-4");
     break;
    default;
System.out.println("Please read the
                    message properly");
```

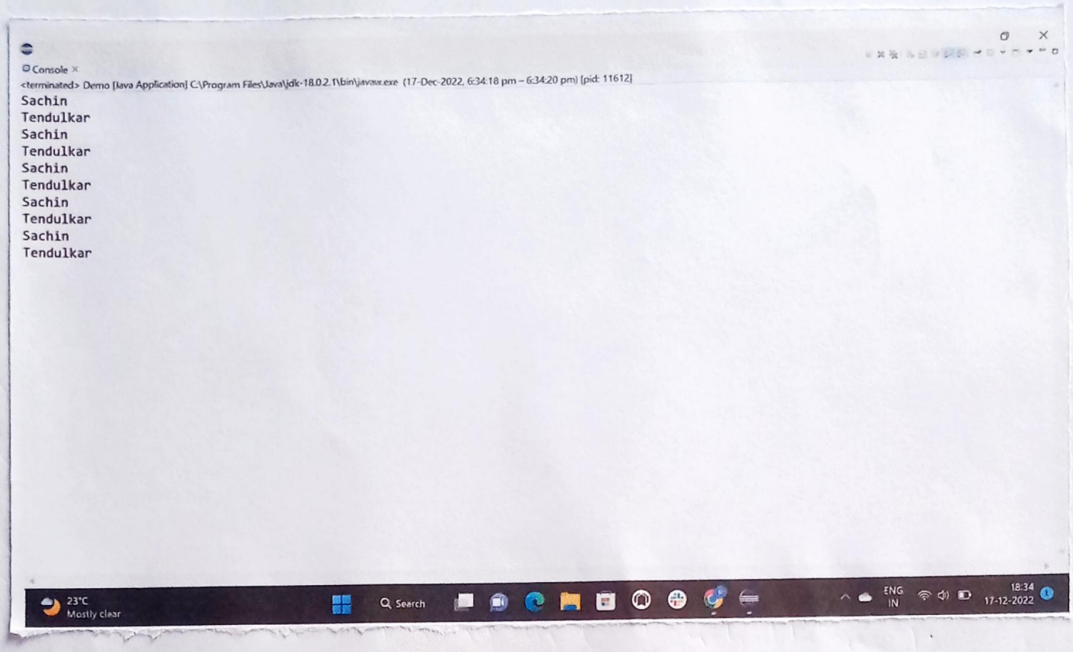
Output :-


Console
Enter Any Number between 1 - 4
3
you have entered 3
Type here to search
ENG
08:53 AM
17-12-2022


Console
Enter Any Number between 1 - 4
1
you have entered 1
Type here to search
ENG
08:53 AM
17-12-2022

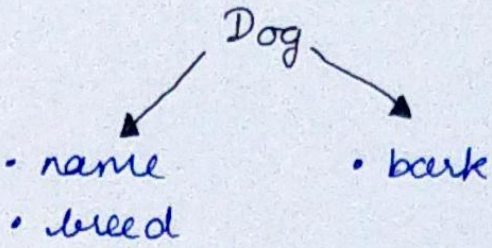
```
class Demo
{
public static void main (String [] args)
{
 System.out.println ("Sachin");
 System.out.println ("Tendulkar");

 System.out.println ("Sachin");
 System.out.println ("Tendulkar");

 System.out.println ("Sachin");
 System.out.println ("Tendulkar");

 System.out.println ("Sachin");
 System.out.println ("Tendulkar");

 System.out.println ("Sachin");
 System.out.println ("Tendulkar");
}
}
```

OUTPUT :—


Console
<terminated> Demo [Java Application] C:\Program Files\Java\jdk-18.0.2.1\bin\javaw.exe (17-Dec-2022, 6:34:18 pm – 6:34:20 pm) [pid: 11612]
Sachin
Tendulkar
Sachin
Tendulkar
Sachin
Tendulkar
Sachin
Tendulkar
Sachin
Tendulkar
23°C
Mostly clear
Search
ENG
IN
18:34
17-12-2022

Object Orientation Programming

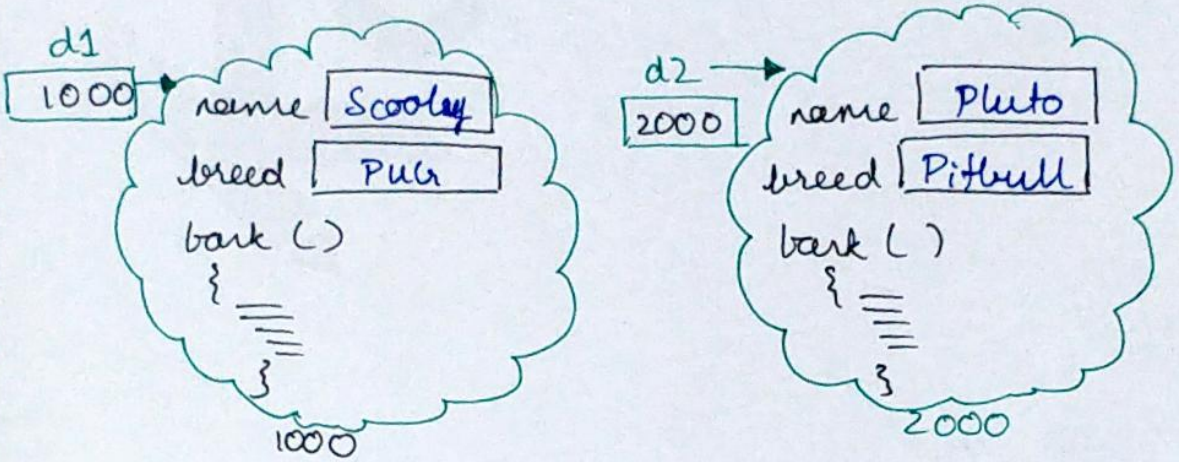
example-2




```
class Dog
{
    String name;
    String breed;
    void bark()
    {
        System.out.println("Bow Bow");
    }
}
class DogApp
{   public static void main(String[] args)
    {
        Dog d1 = new Dog();
        d1.name = "Scooby";
        d1.breed = "pug";
```

```
System.out.println("Details of Dog d1 :-");
System.out.println("Name :"+d1.name);
System.out.println("Breed :"+d1.breed);
d1.bark();

Dog d2 = new Dog();
d2.name = "Pluto";
d2.breed = "pitbull";
System.out.println("Details of Dog d2 :-");
System.out.println("Name :"+d2.name);
System.out.println("Breed :"+d2.breed);
d2.bark();
}
}
```

In heap memory :-



Output :-

For compilation :- javac DogApp.java

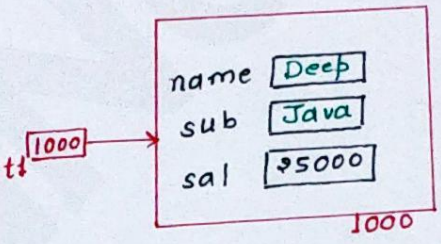
For execution :- java DogApp.

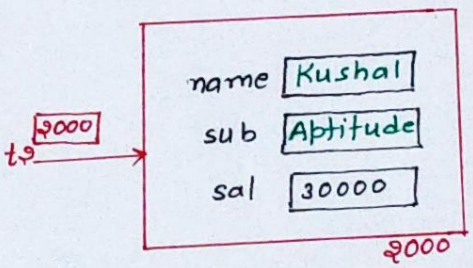
```
Details of Dog d1 :-
Name : Scooby
Breed : Mg
Baw Baw!
Details of Dog d2 :-
Name : Pluto
Breed : pitbull
Baw Baw!
```

Program without Constructor in Java

```
class Teacher
{
 String name;
 String sub;
 int sal;
void teach()
{
 System.out.println("A teacher teaches");
}
}
class TeacherApp
{
public static void main(String[] args)
{
Teacher t1 = new Teacher();
t1.name = "Deep";
t1.sub = "Java";
t1.sal = 25000;

Teacher t2 = new Teacher();
t2.name = "Kushal";
t2.sub = "Aptitude";
t2.sal = 30000;
}
}
```

Instance variable





Constructor in Java

Java provide us much more efficient way for giving the values for our instance variables which are present inside the object and that is called as constructor.

Name of the constructor is always same as the name of the class.

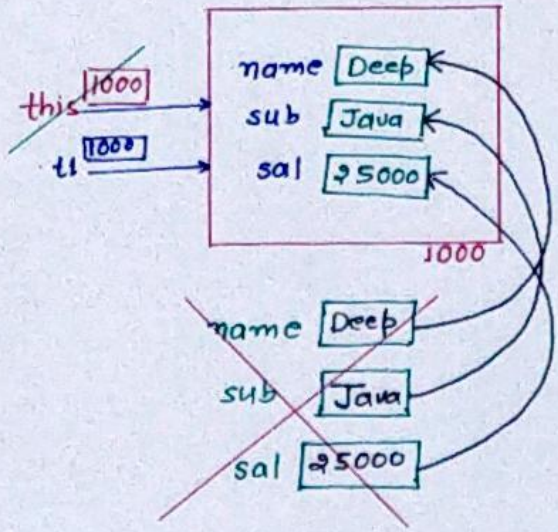
Whatever variable Teacher constructor is accepting are known as local variable.

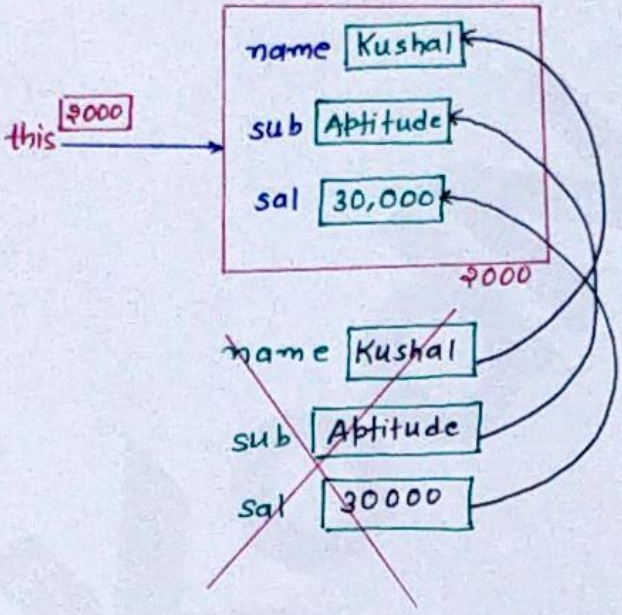
When is constructor going to get called - During the Object creation.

What is the use of a constructor - We can use a constructor to give or set the value for our instance variable.

Program with Constructor in Java

```
class Teacher
{
 String name;
 String sub;
 int sal;
 void teach()
 {
  System.out.println("A teacher teaches");
 }
Teacher(String name, String sub, int sal)   } Local variables
 {
  this.name = name;
  this.sub  = sub;      } Instance variables
  this.sal  = sal;
 }
}
class TeacherApp
{
 public static void main(String[] args)
 {
  Teacher t1 = new Teacher("Deep", "Java", 25000);
  Teacher t2 = new Teacher("Kushal", "Aptitude", 30000);
 }
}
```


this
1000
t1
1000
name
Deep
sub
Java
sal
25000
1000
name
Deep
sub
Java
sal
25000


this
2000
name
Kushal
sub
Aptitude
sal
30,000
2000
name
Kushal
sub
Aptitude
sal
30000

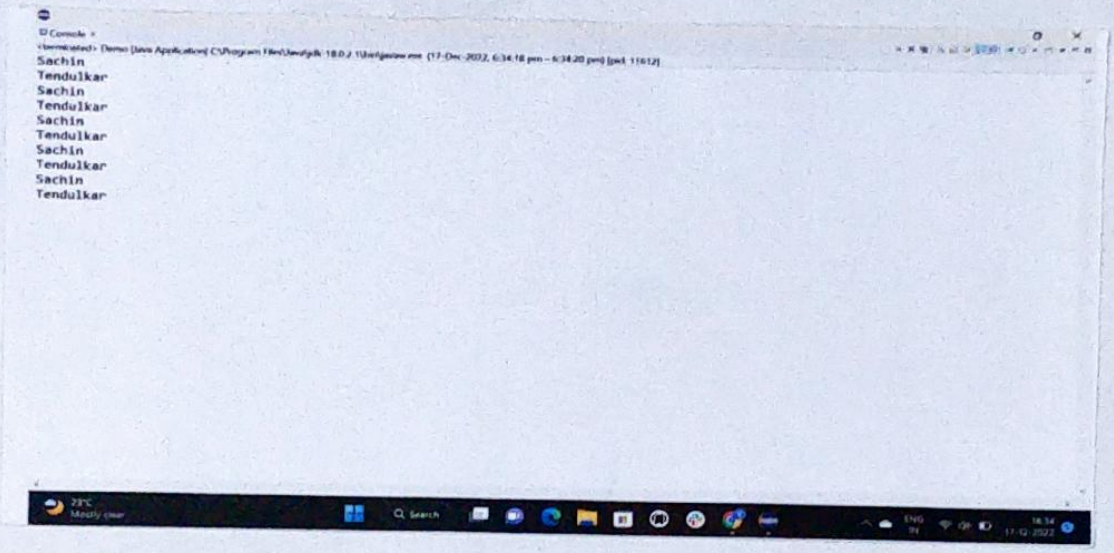
```
Class Demo {
Public static void main (String [] args){
for(int i = 1 ; i<=5; i++)
 {
 System.out.println("Sachin");

 System.out.println ("Tendulkar");
 }
 }
}
```

Output:-

```
Sachin
Tendulkar
Sachin
Tendulkar
Sachin
Tendulkar
Sachin
Tendulkar
Sachin
Tendulkar
```

```
class Demo
{
Public static void main (String [] args)
{
    int i=1;
while (i<=5;)
{
System.out.println ("Sachin");
System.out.println ("Tendulkar");
    i++;
}
}
}
```

output:-



```
class Demo
{
Public static void main (String [] args)
{
    int i=1;
while (i<=5;)
{
System.out.println ("Sachin");
System.out.println ("Tendulkar");
    i++;
}
}
}
```

output:-