```
class EMPLOYEE
{
}
class MANAGER extends EMPLOYEE
{
}
class DIRECTOR extends EMPLOYEE
{
}
class CHECKINSTANCE
{
public void REACT(EMPLOYEE e)
        {
                if(e instanceof MANAGER)
                {
                        System.out.println("THIS IS MANAGER");
                }

                else if(e instanceof DIRECTOR)
                {
                                System.out.println("THIS IS DIRECTOR");
                }
                else
                {
                        System.out.println("This is Employee");
                }
        }
}
```

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

                CHECKINSTANCE ue=new CHECKINSTANCE();

//EMPLOYEE ob=new EMPLOYEE();
        //EMPLOYEE ob=new MANAGER();
                //EMPLOYEE ob=new DIRECTOR();
                ue.REACT(ob);

        }
}
```

Variable Argement(Vaar Args)

It is used to passed number of arguments to function as parameter at runtime.

```
class vaar
{
   int sumArrays(int... intArrays)
   {
      int sum;
   sum=0;
      for(int i:intArrays)
      {

   sum +=i;
  }
```

```
        return(sum);
    }

    public static void main(String args[])
    {
        vaar va = new vaar();
        int sum;

        sum = va.sumArrays(10,20,30,40,50,60,70);
        System.out.println("The sum of the numbers is: " + sum);
    }
}
```

```
class parent
{
int a;
/*parent()
{
System.out.println("Default Construtor Parent");
}*/
parent(int a)
{
this.a=a;
System.out.println("This is Parameterised Constructor"+a);
}
}
class child extends parent
{
```

```
child()
{
//super();
super(5);
System.out.println("This is child class constructor");

}
public static void main(String[]args){
child i=new child();
}
}
```

```
class equality
{
public static void main(String[]args)
{
//String a="hello";
//String b="hello";
String a=new String("hello");
String b=new String("hello");
if(a==b)
//if(a.equals(b))
{
System.out.println("Equals");
}
else
{
System.out.println("Not Equals");
}
```

```
}
}
```

```
public class WrapperExample1{
public static void main(String args[])
{
//Converting int into Integer
int a=20;
Integer i=new Integer(a);

int j=i.intValue();

System.out.println("Base valuee"+a+""+"Boxing "+i+""+"Unboxing "+j);
}
}
```

Primitive Type	Wrapper class
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

Wrapper class Example: Primitive to Wrapper

```
public class WrapperExample1
{
public static void main(String args[]){
//Converting int into Integer
int a=20;
Integer i=Integer.valueOf(a);//converting int into Int
eger
Integer j=a;//autoboxing, now compiler will write Int
eger.valueOf(a) internally

System.out.println(a+" "+i+" "+j);
}}
```

Wrapper class Example: Wrapper to Primitive

```
public class WrapperExample2{
public static void main(String args[]){
//Converting Integer to int
Integer a=new Integer(3);
int i=a.intValue();//converting Integer to int
int j=a;//unboxing, now compiler will write a.intValu
e() internally

System.out.println(a+" "+i+" "+j);
}}
```

---

class Bike9{

```
 final int speedlimit=90;//final variable
 void run(){
  speedlimit=400;
 }
 public static void main(String args[]){
 Bike9 obj=new  Bike9();
 obj.run();
 }
       }//end
```

```
class Bike{
  final void run()
{
System.out.println("running");
}
}

class Honda extends Bike{
   void run()
{System.out.println("running safely with 100kmph");
}

   public static void main(String args[]){
   Honda honda= new Honda();
   honda.run();
   }
       }
```

```
final class Bike
{
}

class Honda1 extends Bike{
  void run()
{
System.out.println("running safely with 100kmph");
}
```

```
  public static void main(String args[]){
  Honda1 honda= new Honda();
  honda.run();
  }
}
```
