SRI VENKATESA PERUMAL
COLLEGE OF ENGINEERING AND TECHNOLOGY
|AUTONOMOUS|ACCREDITED BY NAAC|
RVS Nagar, K.N. Road, Puttur, Chittoor dist, AP.|www.svpcet.org

Department of ECE

Subject : oop's using java **Subject code**:200E0501

Class: III YEAR I SEM

<u>OBJECT IN CLASS</u>:

* A special class defind by object,other classes follows sub classes of objects.

*That is a object is the super classes of associated to the other classes.

*If a reference variable of object that will refer other classes of objects.

*A arrays must be implemented in a class then the variable type of an object must refers an arrays.

*Varies kinds of methods to follows obeject.

Method	Meaning
Object clone()	To ceate an object having future that will same future occurs in other object it refers cloning
Boolean equql(object object)	It determines the object which equal to another object.
Void finalized()	It called before the un used objects must recycled.
Class getclass()	To determine a class all objects occur in run time.
Int hash code()	It returns hash code which will invoking all set of objects.
Void notify()	It resume execution and a threads must waiting and it will invoking all Set of objects.
Void notify all()	It resumes the execution of all threads must waiting. henceit will also invoking all set of objects.
void wait()	It represents wait for thread execution.
string to string()	It obtains the Shing which describes all set of objects.

NOTE:

*In above methods get class notify,notify wait all declared as final keyword.

*Equal and two string methods which will compared two kinds of objects one object is true and another object is false.

*A string declares an object which provides discription of object that will called.

INHERITANCE:

A inheritance is a class stone of object oriented programming and it is implemented hierarchical representation. If we are adding a class and it is maintains desired set of elements.if we are adding two or more classes than all classes must be unique manner. If a class is inheritanted then refers as super class and a class will inheritance then it refers as Subclass. A Super is always visionary of subclasses. A super class represents Various variables and instantenous of elements in specific programs. for example a class will incorperate to another class which will usea keyword name called extent.

Example of Inheritance:

```
class A & class B class A{
    int i,j;
    void show i,j(){
    system.out.println("i and j"+i+" "+j);
    }
    }

    subclass B extends A{
    int k;
    void show k(){
    system.out.println("K "+k);
    }
    void sum(){
    system.out.println("i+j+k"+(i+j+k));
    }
```

Program:

Simple Inheritance of Java Program

```
class simple inheritance{
public static void main(string[] args){
A supob=new A();
B supob=new B();
system.out.println("contents of supobj");
supob.showij();
system.out.println();
sub ob.i=7;
sub ob.j=8;
sub ob.k=9;
system.out.println("i,j and k are sum of sub ob",(i+j+k));
}
}
```

```
output:
       contents of supob:
         i=10
         j=20
       content of supob:
         i=7
         j=8
         k=9
       The sum of i,j,k in subob=24
```

Note:

In above program super classes can be used itself and Subclasses will access the members of superclass as a variables i & j which is called methods. Java doesn't Supports many super classes and single suubclass, it will supports many sub classes and single Super Class.

CLASS:

*A class defines group and order of elements.

*A Class is also defines a a new type.

*A new type of classes then it Will Create an objects.

*A object is instances of classes.

*A class is a template of objects.

*Generally a class must combines objects and Instances, those Properties are interchangeable.

The general form of class:

```
class classname {
                    Type  instance variable 1
                     .    .    .
                     .    .    .
                     .    .    .
                     .    .    .

                    Type  instance  variable N
                    Type method 1(parameter list)
                     .   .   .   .
                     .   .   .   .
                     .   .   .   .
                     .   .   .   .

                    Type method N(parameter list)
                    }
```

In above general form of class.a class declares Class name.a combination of data & variable is Called instance variable. All code must present in the method itself. A combination of method and variable is Called members. A classes to do not Create objects than it is not called main declaration. if a class declares and Creates one or more objects than it is Called main decleration.

Program

```
class Box{
double weight;
double height;
double depth;
class BoxDemo{
public static void main(string[] args){
Box mybox=new Box();
double vol;
my box weight=10;
my box height=15;
my box depth=20;
vol=mybox.weight*mybox.height*myboxdepth;
system.out.println("Vol is :"+vol);


output:
     Vol is 3000
```

DECLARING OBJECT:

*A declaring represents a known facts and Declaring Objects represents a class must creating data types that types will Create objects.

*A declaring objects follows to important reason.the first reason is A Variable can't define the object but varible can refer the objects.

*The Second reason represents a object to assign a new variable it will apply a new operator called new" which allocates a memory both object and reference.

*A reference maintains less on high memory between returns and reference value. hence a reference is also stores variables.

*A object of Classes wil dynamically allocated.

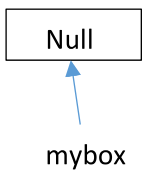
*A declaring object defines two statements as follows the first statement is box mybox; this statement represents declaring a object through reference and second statement is mybox=new box(); that statement represents to allocates the object in to the reference.

*A first statement will plan to execute and it shows compile time error and it maintains NULL value.

*The Second Statement will plan to execute than the object will store memory address about the object,the statement will execute and it doesnot Show any compaile time error.

Statement **Effect**

Box mybox;



Mybox=newBox();

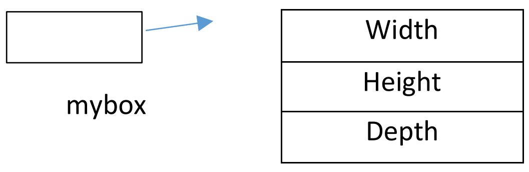


Fig: Declaring object with type is Box.

Method:

* A Method in a java is the way of made it a class follows a method. it maintains two properties which is instances & method. A instance is a demand & Suggestion of classes and method is a flexiable & Powerful Category between object classes.

*The general form of method as follows

type hame (Parameter list)

*In hame above General form the type.. represents a data methods, a name define and Specifies a method Name and a Parameter list defines a sequence type and identifiers of methods which will returns the values

Ex: To demonstrate a Java program using method?

```
class Box{
double width;
double height;
double depth;
}
vois vol(){
system.out.println("Vol is "+vol());
system.out.println("Width*height*depth);
class BoxDemo{
public static void main(string[] args){
Box mybox1=new Box();
Box mybox2=new Box();
mybox1.width=10;
mybox1.height=20;
mybox1.depth=15;
mybox2.width=3;
mybox2.height=6;
mybox2.depth=9;
mybox1.vol();
mybox2.vol();

}
}
}

output:
        3000
        162
```

Constructor:

*Basically a constructor define two constructi building and making an elements.

*A Constructor is also defines, which is initialize all different kinds of Variables then every time create an instances.

*because the requirement of initialization is most common thing and also Java will allows objects then it must specifies to Create instances and it is atomatically initializ through the use of constructor.

*A constructer also defines which is initalize objects and it is also create different kinds of instances. In the Same Way of procedure classes also access similar methods which is syntatic representation.

* once the Constructor is defined it automatically immedately of object before & new operators are assigned which is complted by an constructor.

EX:

Program

```
class Box{
double width;
double height;
double depth;
system.out.println("contructor is Box:");
width=10;
height=10;
depth=10;
void volume(){
return widht*height*depth;
}
}
class BoxDemo{
public static void main(string args[]);
{
Box mybox1=new Box();
Box mybox2=new Box();
double vol;
vol=mybox1.volume();
system.out.println("vol is:"+vol);
vol=mybox2.volume();
system.out.println("vol is"+vol);

}
}

output:
    constructor of Box:
    volume is:1000
    volume is:1000
```

<u>ARGUEMENT PASSING:</u>

*A Arguement passing represents a statement, that Statement willeffects programmer and subrotines.

*A Argument passing can be categories in two ways they are:

   *call by reference

   *call by value

*Call by by Value defines it Copies the values from Argument into parameter of Subroutines. if parameter will change there is no effect from arguements than it is consider as call by value.

*Call by reference defines a parameter must occupies the variable but it does not copy the values. Hence the parameter is also change.it will effect from all kinds of arguments. A Sub routine is also consider as call by reference which will maintain actual parameter with various kinds of objects.

program:

DEMONSTRATE A JAVA PROGRAM BY USING CALL BY VALUE:

```
class Test{
void int(i,j){
i*=2;
j/=2;
}
}
class call_by_value{
public static void main(string args[]){
Test ob =new Test();
int a=15,b=20;
system.out.println("a and b before call:"+a+" "+b);
ob.meth(a,b);
system.out.println("a and b after call:"+a+" "+b);

}
}

output:
        a and b before call 15 20
        a and b after call 30 10
```

NOTE:

*In the above code a operation of method name call meth()

*It has no effect the values then all primitive types must pass themethods than it reference has call by value.

Program

DEMONSTRATE A JAVA PROGRAM BY USING CALL BY REFERENCE.

```
class Test{
int a,b;
Test(int i,int j){
a=i;
b=j;
}
void meth(Test 0){
o.a*=2;
o.b/=2;
}
}

class call_by_reference{
public static void main(string args[]){
Test ob =new Test(15,20);
system.out.println("ob.a and ob.b before call:"+ob.a+" "+ob.b);
ob.meth(ob);
system.out.println("ob.a and ob.b after call:"+ob.a+" "+ob.b);

}
}

output:
        ob.a and ob.b before call 15 20
        ob.a and ob.b after call  30 10
```

Note:

In above Source code the operation & effects which called a method meth() and it effects the values must be change through the arguments and implicit kind of an Object must be passed through the method then it refers as call by reference.

Return objects:

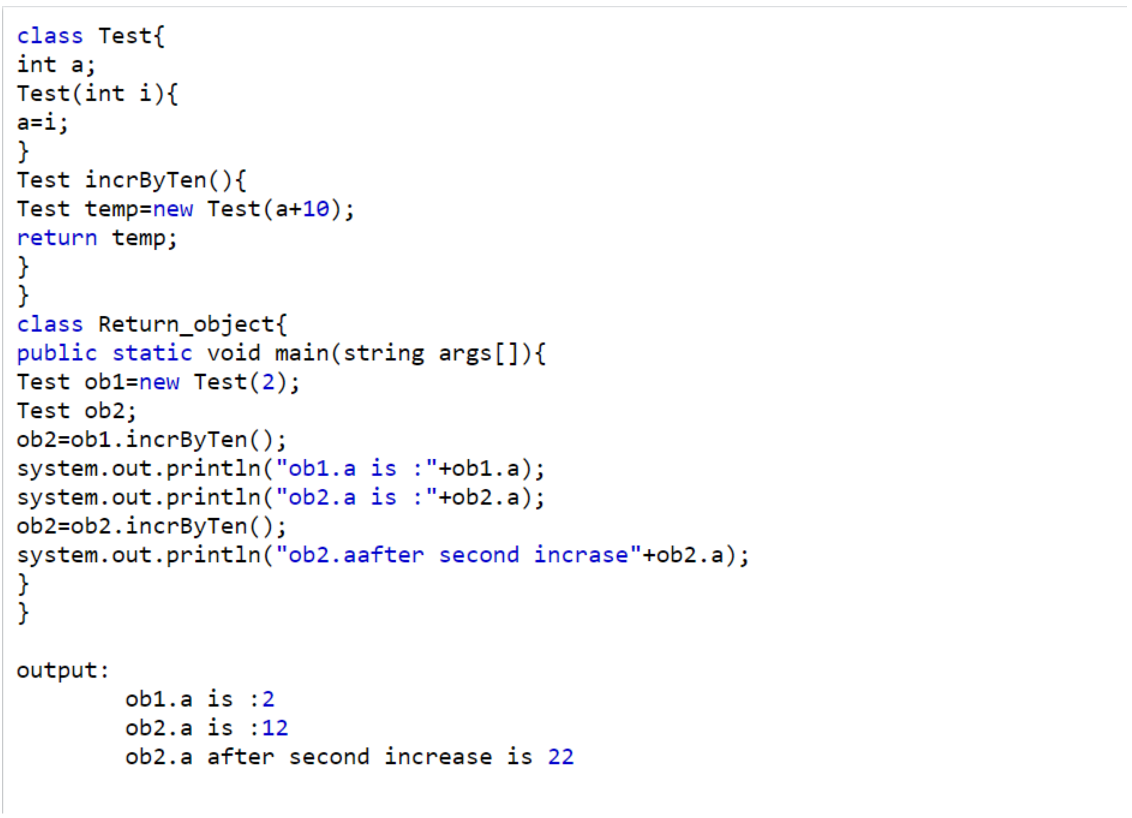
* A returning defines come back state & object defines Purpose & things.

* A return object defines any state of methods must written any type of data & it will return objects.

* The following Program used a method called iner By Ten(), that method returns a object which will use the Variable 'a' & it is greater than 10. which will invoking the objects.

Ex:

Program



```
class Test{
int a;
Test(int i){
a=i;
}
Test incrByTen(){
Test temp=new Test(a+10);
return temp;
}
}
class Return_object{
public static void main(string args[]){
Test ob1=new Test(2);
Test ob2;
ob2=ob1.incrByTen();
system.out.println("ob1.a is :"+ob1.a);
system.out.println("ob2.a is :"+ob2.a);
ob2=ob2.incrByTen();
system.out.println("ob2.aafter second incrase"+ob2.a);
}
}

output:
        ob1.a is :2
        ob2.a is :12
        ob2.a after second increase is 22
```

Recursion:

* A recursion defines it happens and it occurs.

* Java Supports recursion.

*A Recursion is the process and it shows something and it call itself. Then it is called recursion.

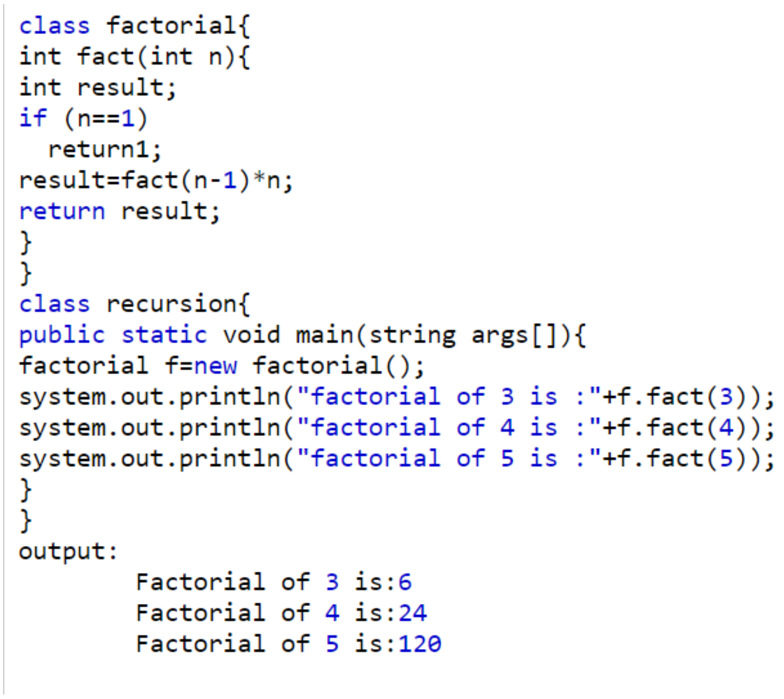
* As it Relates of Java programming, a recursion must allows specific kind of attributes. Things then it. which is called by methods. Then it is called itself.

*A method must be called itself refers as recursive.

*for example the best suitable recursion is factorial of a number. A factorial of a number which will product, the whole numbers from 1 to N. If it is the factorial of thee and it formed as 3 = 3x2x1 = 6. The following Program to find factorial using recursion.

Program

Write a java program to find factorial using recursion.



```
class factorial{
int fact(int n){
int result;
if (n==1)
  return1;
result=fact(n-1)*n;
return result;
}
}
class recursion{
public static void main(string args[]){
factorial f=new factorial();
system.out.println("factorial of 3 is :"+f.fact(3));
system.out.println("factorial of 4 is :"+f.fact(4));
system.out.println("factorial of 5 is :"+f.fact(5));
}
}
output:
        Factorial of 3 is:6
        Factorial of 4 is:24
        Factorial of 5 is:120
```

Note:

 * In above source code an operation to made a fact method which called an argument and it returns 1. again the argument is called as Fact (n-1)*n

 *Hence fact Called (n-1) with specifies to repeat N' No. of times follows a method & returning no. of times

<u>THIS:</u>

* It is a keyword which defines actual Points of object.

*It is also defines to refer different methods & objects.

*Once it refers the methods & objects which Will Permetable.

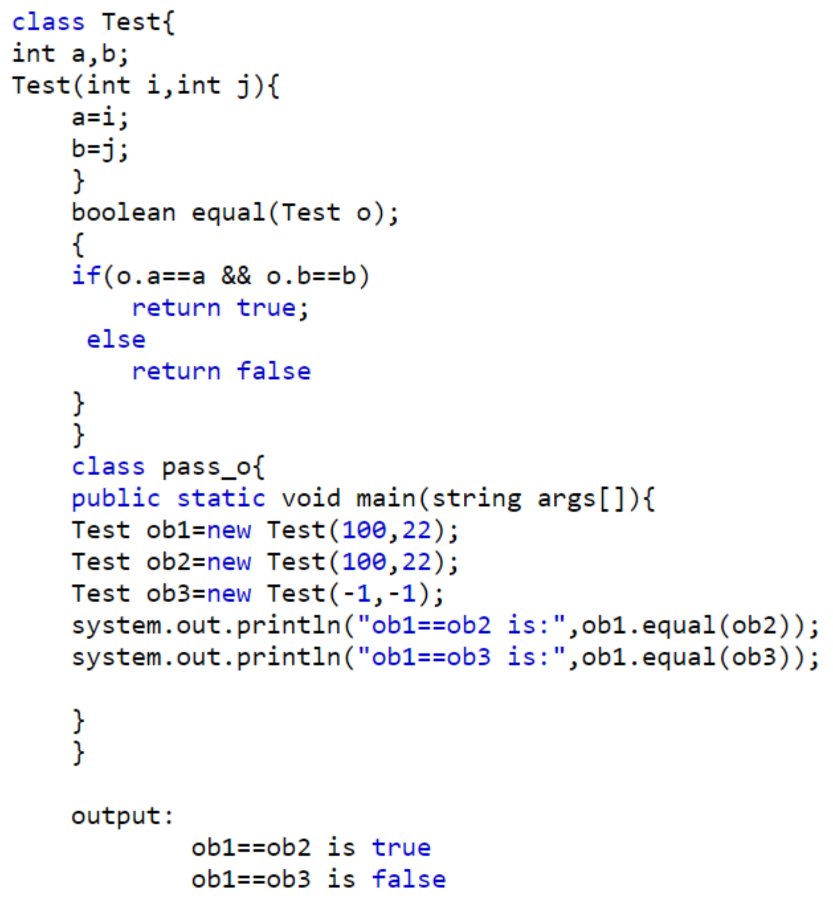
*A redundancy occurs this keyword hence it is Valid Correct.

* A Valid set of retundancy, must invoking objects & methods.

A Invoking in this keyword represents, Something called a method & reteres. The General form of This keyword as follows.

```
Box(height h,width w,depth d){
                    This.height=h;
                    This.width=w;
                    This.depth=d;
                    }
```

<u>Object as Parameters:</u>



```
class Test{
int a,b;
Test(int i,int j){
    a=i;
    b=j;
    }
    boolean equal(Test o);
    {
    if(o.a==a && o.b==b)
        return true;
     else
        return false
    }
    }
    class pass_o{
    public static void main(string args[]){
    Test ob1=new Test(100,22);
    Test ob2=new Test(100,22);
    Test ob3=new Test(-1,-1);
    system.out.println("ob1==ob2 is:",ob1.equal(ob2));
    system.out.println("ob1==ob3 is:",ob1.equal(ob3));

    }
    }

    output:
            ob1==ob2 is true
            ob1==ob3 is false
```

NOTE:

*In above representation a object has parameter which represents Character maintains member which follows a system.

*It is also defines Passing a method•

*It's command and correctstate need to pass objects and methods•

*In above programme a equal represents a method and test is a class name.

*if a method compare two different objects which is invoked ajnd is passed.

*if two different objects are equal then it shows true.if two different objects are not same then it shows false.

*hence a class name represents(or)java build in types.