//Program-1 Call by value

```
class Tester {
    public static void main(String[] args) {
        int a = 30;
        int b = 45;
        System.out.println("Before swapping, a = " + a + " and b = " + b); // Invoke the
        swap  method

        swapFunction(a, b); //main is a static method, hence it can call only static
    method

        System.out.println("\n**Now, Before and After swapping values will be same
    here**:");
        System.out.println("After swapping, a = " + a + " and b is " + b);
    }
    static void swapFunction(int a, int b) {
        System.out.println("Before swapping(Inside), a = " + a + " b = " + b); // Swap n1
    with n2
        int c = a;
        a = b;
        b = c;
        System.out.println("After swapping(Inside), a = " + a + " b = " + b);
    }
}
//if static void swapFunction(int,int) is in another class, call it by className.swapFunction(a,b);
```

// Program-2 Creating reference of an object

```
public class Main{

  public static void main(String[] args) {
    Point p1 = new Point();
    p1.x=10;
    p1.y=20;
    p1.z=30;

    System.out.println(p1. calDistanceOrigin ());
    Point p2=new Point();
    p2.setData(5,15,25);

    System.out.println(p2. calDistanceOrigin ());
    p2.translate(5, 5, 5);
    System.out.println(p2.x+" "+p2.y+" "+p2.z);

     Point p3=p1; //Does it call constructor? --> No
    p3.x=50;
    System.out.println(p1.x); // Does p3.x=50 change p1.x? --> Yes
  }
}
class Point{
  double x,y,z;
  Point(){
        x=y=z=0;
        System.out.println("In default constructor");
  }
  Point(int ix, int iy, int iz){
        x=ix;
        y=iy;
        z=iz;
        System.out.println("In parameterized constructor");
  }
  void setData(int a, int b, int c){
    x=a;
    y=b;
    z=c;
  }
  double calDistanceOrigin (){
    return Math.sqrt(x * x + y * y + z*z);
  }
  void translate(int dx, int dy, int dz){
    x = x + dx;
    y = y + dy;
    z = z + dz;
  }
  void printData()
  {
        System.out.print("Point is:");
        System.out.println("("+ x + ", " + y + ", " + z + ")");
  }

}
```

//Program-3 Call by reference

```
class Tester {
    int a, b;
    Tester(){
        System.out.println("In default constructor”);
    }
    public static void main(String[] args) {
        Scanner s=new Scanner(System.in);
        Tester t = new Tester();

        System.out.println("enter a and b");
        t.a=s.nextInt();
        t.b=s.nextInt();

        System.out.println("Before swapping"); // Invoke the swap method
      t.printData();
      t.swapFunction(t);
      System.out.println("After swapping");
      t.printData();
    }
    void swapFunction(Tester t) { //Will it call default constructor?
        // swap a with b
        int c = t.a;
        t.a = t.b;
        t.b = c;
    }
    void printData(){
      System.out.println("a ="+a+" ,b="+b);
    }
}
```

//Program-4 Returning objects

Program to implement returning objects.

```
class Rectangle {
    int length;
    int breadth;
    Rectangle(){
        System.out.println("In default constructor”);
    }
    Rectangle(int l,int b) {
        System.out.println("In parameterized constructor”);
        length = l;
        breadth = b;
    }
    Rectangle getRectangleObject() {
        Rectangle rect = new Rectangle(10,20);
        return rect; //Will it call default constructor?
    }
}
class RetOb {
    public static void main(String args[]) {
        Rectangle ob1 = new Rectangle(40,50);
        Rectangle ob2; //Will it call default constructor?
        ob2 = ob1.getRectangleObject();
        System.out.println("ob1.length : " + ob1.length);
        System.out.println("ob1.breadth: " + ob1.breadth);
        System.out.println("ob2.length : " + ob2.length);
        System.out.println("ob2.breadth: " + ob2.breadth);
    }
}
```