B)Define a class named Shape, make objects of Shape class like square,rectangle,triangle,circle overload the constructors to create the objects. Use calcarean() method to calculate areas of each of the objects by using method overloading. Input the data member by command line arguments.

```
/*method overloading(rectangle,squre,triangle,circle)*/
class Shape
{
 int length;
 int breadth;
 float base;
 float height;
 float radious;
 Shape(int a,int b) //assuming rectangle
  {
    length=a;
    breadth=b;
  }
 Shape(int a)     //assuming square
  {
    length=breadth=a;
  }
 Shape(float a,float b)  //triangle
  {
    base=a;
    height=b;
  }
 Shape(float a)     //circle
  {
```

```
    radious=a;
  }
 int area()
  {
    return(length*breadth);
  }
 float area_t()
  {
    return(.5f*base*height);
  }
 float calcArea()
  {
    return(3.14f*radious*radious);
  }
}
class Overloading_srtc
 {
   public static void main(String args[])
    {
      int i=Integer.parseInt(args[0]);
      int j=Integer.parseInt(args[1]);
      int k=Integer.parseInt(args[2]);
      float p=Float.parseFloat(args[3]);
      float q=Float.parseFloat(args[4]);
      float m=Float.parseFloat(args[5]);
      Shape r1=new Shape(i,j);
      Shape r2=new Shape(k);
      Shape r3=new Shape(p,q);
      Shape r4=new Shape(m);
      System.out.println("area of r1 is: "+r1.area());
      System.out.println("area of r2 is: "+r2.area());
```

```
        System.out.println("area of r3 is: "+r3.area_t());
        System.out.println("area of r4 is: "+r4.calcArea());
    }
}
```