Practical 4

Write a program based on method and constructor overloading.

import java.util.\*;

class AreaClass {

  int side1;

  int side2;

  int side3;

  int side4;

  public AreaClass(int a, int b, int c) {

    side1 = a;

    side2 = b;

    side3 = c;

    System.out.println(

      &quot;It is Triangle with sides as \n&quot; +

      a +

      &quot; units,&quot; +

      b +

      &quot; units,&quot; +

      c +

      &quot; units&quot;

    );

  }

  public AreaClass(int a) {

    side1 = a;

    System.out.println(&quot;It is a Square with side as \n&quot; + a + &quot; units&quot;);

  }

  public AreaClass(int a, int b) {

    side1 = a;

    side2 = b;

    System.out.println(

      &quot;It is Rectangle with side as \n&quot; + a + &quot; units,&quot; + b + &quot; units.&quot;

    );

  }

  void getArea(int a) {

    double area = a \* a;

    System.out.println(&quot;Area of Square is &quot; + area + &quot; sq.units&quot;);

  }

  void getArea(double base, double side) {

    double area = 0.5 \* base \* (side \* 1.73) / 2;

    System.out.println(&quot;Area of Triangle is &quot; + area + &quot; sq.units&quot;);

  }

  void getArea(int a, int b) {

    double area = a \* b;

    System.out.println(&quot;Area of Rectangle is &quot; + area + &quot; sq.units&quot;);

  }

}

public class exp4{

    public static void main(String args[]){

        AreaClass shape1 = new AreaClass(5);

        shape1.getArea(5);

        System.out.println();

        AreaClass shape2 = new AreaClass(5,6);

        shape2.getArea(5,6);

        System.out.println();

        AreaClass shape3 = new AreaClass(20, 20,20);

        shape3.getArea(20, 20);

    }

}

