**Java Lab Sheet 4**

**-Mohith LS  
AM.SC.U3CSC21036**

1. Create two classes, one parent class and one child class both are having same method with different definitions...

public class L4Q1

{

    public static void main(String args[])

    {

        Child c = new Child();

        Parent p = new Child();

        c.print();

        p.print();

    }

}

class Parent

{

    public void print()

    {

        System.out.println("Parent");

    }

}

class Child extends Parent

{

    public void print()

    {

        System.out.println("Child");

    }

}

Output:



2. Create shape class with three different definitions for the method area...

import java.util.Scanner;

import java.math;

public class L4Q2

{

    public static void main(String args[])

    {

        Scanner sc = new Scanner(System.in);

        Shape myshape = new Shape();

        System.out.print("Enter side of square: ");

        double side = sc.nextDouble();

        System.out.println("Area of square: " + myshape.area(side));

        System.out.print("Enter sides of rectangle: ");

        double length = sc.nextDouble();

        double breadth = sc.nextDouble();

        System.out.println("Area of rectangle: " + myshape.area(length, breadth));

        System.out.print("Enter sides of triangle: ");

        double a = sc.nextDouble();

        double b = sc.nextDouble();

        double c = sc.nextDouble();

        System.out.println("Area of triangle: " + myshape.area(a, b, c));

    }

}

class Shape

{

    public double area(double a)

    {

        return a\*a;

    }

    public double area(double l, double b)

    {

        return l\*b;

    }

    public double area(double a, double b, double c)

    {

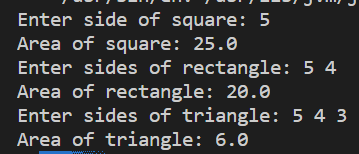
        double s = (a+b+c)/2;

        return Math.sqrt(s\*(s-a)\*(s-b)\*(s-c));

    }

}

Output:



3. Create an interface ‘polygon’ and...

import java.util.Scanner;

interface Polygon

{

    public void getArea();

}

class Square implements Polygon

{

    private double side;

    public Square(double side)

    {

        this.side = side;

    }

    public void getArea()

    {

        System.out.println("Area of square: " + side\*side);

    }

}

class Rectangle implements Polygon

{

    private double length;

    private double breadth;

    public Rectangle(double length, double breadth)

    {

        this.length = length;

        this.breadth = breadth;

    }

    public void getArea()

    {

        System.out.println("Area of rectangle: " + length\*breadth);

    }

}

public class L4Q3

{

    public static void main(String args[])

    {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter side of square: ");

        double side = sc.nextDouble();

        Square s = new Square(side);

        s.getArea();

        System.out.print("Enter sides of rectangle: ");

        double length = sc.nextDouble();

        double breadth = sc.nextDouble();

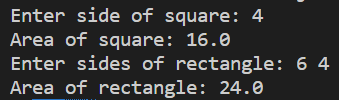
        Rectangle r = new Rectangle(length, breadth);

        r.getArea();

    }

}

Output:



4. Write a Java program to show the use of abstract class and abstract method.

//Testing abstract class and method

abstract class myclass

{

    abstract void mymethod();

}

class myclass2 extends myclass

{

    void mymethod()

    {

        System.out.println("This is mymethod");

    }

}

public class L4Q4

{

    public static void main(String args[])

    {

        myclass2 m = new myclass2();

        m.mymethod();

    }

}

Output:

