Q1. Create a class called Person with attributes such as name and age. Derive a class called Student from Person that adds an attribute student ID. Write a program to demonstrate single inheritance by creating objects of both classes and displaying their attributes?

```
package TSGOL.com;

public class Person {

protected String pname; protected int age; //Constructor

public Person(String name, int age) {

this.pname=name;

this.age=age;

}

public void display() {

System.out.println("Name: " + name);

System.out.println("Age: " + age);

}

} public class Student extends Person { private String
studentId;

//Constructor

public Student(String name, int age, String studentId) {
super(name, age);//Calls the constructor of the superclass
this.studentId = studentId;

}

public void display() {

super.display(););//Driven by super class method
System.out.println("Student id: " + studentId);

}

public static void main(String[] args) {
```

```
Person p = new Student("SAIKUMAR", 24, "AF0311766");//Creates
an object of

Student class and assigns it to a Person class reference
variable

p.display();//Calls the display method of the object

}

}
```

Output :

```
Name: SAIKUMAR

Age: 24

Student id: AF0311766
```

Q2. Design a class called Shape with methods to calculate the area and perimeter. Derive classes like Circle, Rectangle, and Triangle from Shape. Write a program to create objects of these classes and compute their areas and perimeters?

```
package TSGOL.com;

abstract class Shape

{

abstract double Area();//Abstract method for calculating the
area

abstract double Perimeter();//Abstract method for calculating
the perimeter

}

public class Circle extends Shape

{

private double radius;

//Constructor

public Circle(double radius)
```

```
{
this.radius=radius;
}
//Implementation of abstract method
double Area()
{
return Math.PI*radius*radius;
}
// Implementation of abstract method
double Perimeter()
{
return Math.PI*radius;
}
}
public class Rectangle extends Shape
{
private double length;
private double width;
//Constructor
public Rectangle(double length, double width)
{
this.length = length;
this.width = width;
}
// Implementation of abstract method
```

```
double Area()

{

return length*width;

}

// Implementation of abstract method double

Perimeter()

{

return 2*(length*width);

}

}

public class Triangle extends Shape

{

private double side1;

private double side2;

private double side3;

//Constructor

public Triangle(double side1, double side2, double side3)

{

this.side1 = side1;

this.side2 = side2;

this.side3 = side3;

}

// Implementation of abstract method

double Area()

{
```

```
double s = (side1 + side2 + side3) / 2; // calculate semiperimeter

return Math.sqrt(s * (s - side1) * (s - side2) * (s - side3));

}

// Implementation of abstract method

double Perimeter()

{

return side1 + side2 + side3;

}

}

public class ShapeSimulation

{

public static void main(String []a)

{

Shape c,r,t;//Objects variables created by use Shape class name

c = new Circle(2); //Creates a new Circle object with a values

r = new Rectangle(2, 1.5); // Creates a new Rectangle object with a values

t = new Triangle(3, 3, 3); // Creates a new Triangle object with a values

System.out.println("Area of the Circle = "+c.Area());

System.out.println("Perimeter of the Circle = "+c.Perimeter());

System.out.println("-------------------------------------------------------------------");

System.out.println("Area of the Rectangle = "+r.Area());

System.out.println("Perimeter of the Rectangle = "+r.Perimeter());
System.out.println("------------------------------------------
--------------------------");

System.out.println("Area of the Triangle = "+t.Area());

System.out.println("Perimeter of the Triangle =
"+t.Perimeter());

}

}
```

Output :

```
Area of the Circle = 12.566370614359172

Perimeter of the Circle = 6.283185307179586

--------------------------------------------------------------

Area of the Rectangle = 3.0

Perimeter of the Rectangle = 6.0

--------------------------------------------------------------

Area of the Triangle = 3.897114317029974

Perimeter of the Triangle = 9.0
```

Q3. Create a base class called Animal with a method named sound(), which prints "Animal makes a sound." Derive classes Cat and Dog from Animal. Override the sound() method in each derived class to print "Cat meows" and "Dog barks" respectively. Write a program to demonstrate method overriding by creating objects of the derived classes and calling the sound() method.

```
package TSGOL.com;

public class Animals

{

public void Sound()

{

System.out.println("Animal makes a sound.");
```

```
}

}

public class Cat extends Animals

{

//Overriding method

public void Sound()

{

System.out.println("Cat meows.");

}

}

public class Dog extends Animals

{

//Overriding method

public void Sound()

{

System.out.println("Dog barks.");

}

}

public class AnimalSimulation

{

public static void main(String[] args)

{

Animals a,c, d;// Objects variables created by use Animals
class name

a = new Animals();//Creates a new object of the Animals

c = new Cat();//Creates a new object of the Cat
```

```
d = new Dog();//Creates a new object of the Dog .

a.Sound();//method calling form Animals class

c.Sound();//method calling form Cat class

d.Sound();//method calling form Dog class

}

}
```

Output :

```
Animal makes a sound.

Cat meows.

Dog barks
```

Q4. Design a class called Shape with a method named calculate Area (). Derive classes such as Circle, Rectangle, and Triangle from Shape and override the calculate Area () method in each derived class to compute the area specific to that shape. Write a program to create objects of these classes and invoke the calculate Area () method to calculate and display their respective areas?

```
package TSGOL.com;

abstract class Shape

{

abstract double CaculateArea();

}

public class Circle extends Shape

{

private double radius;

//Constructor

public Circle(double radius)

{
```

```
this.radius=radius;

}

//Overriding method

double CaculateArea()

{

return Math.PI*radius*radius;

}

}

public class Rectangle extends Shape

{

private double length;

private double width;

//Constructor

public Rectangle(double length, double width)

{

this.length = length;

this.width = width;

}

//Overriding method

double CaculateArea()

{

return length*width;

}

}

public class Triangle extends Shape
```

```
{

private double side1;

private double side2;

private double side3;

//Constructor

public Triangle(double side1, double side2, double side3)

{

this.side1 = side1;

this.side2 = side2;

this.side3 = side3;

}

//Overriding method

double CaculateArea()

{

double s = (side1 + side2 + side3) / 2; // calculate semi-perimeter

return Math.sqrt(s * (s - side1) * (s - side2) * (s - side3));

}

}

public class ShapeSimulation

{

public static void main(String []a)

{

Shape c,r,t; //Objects variables created by use Shape class name

c = new Circle(2); //Creates a new object of the Circle with values
r = new Rectangle(2, 1.5); // Creates a new object of the
Rectangle with values

t = new Triangle(3, 3, 3); // Creates a new object of the
Triangle with values

System.out.println("Area of the Rectangle = "+r.Area());

System.out.println("Area of the Rectangle = "+r.Area());

System.out.println("Area of the Triangle = "+t.Area());

}

}
```

Output :

```
Area of the Circle = 12.566370614359172

Area of the Rectangle = 3.0

Area of the Triangle = 3.897114317029974
```