```
import java.util.*;

public class App {
   static Scanner scanner = new Scanner(System.in);
   static Scanner scannerBool = new Scanner(System.in);
   static Scanner scannerStr = new Scanner(System.in);
   public static void main(String[] args) throws Exception {
      GeometricObject triangle = new Triangle(3,4,5,"Red",true);
      displayObject(triangle);
      System.out.println();

      GeometricObject rectangle = new Rectangle(5,4,"Blue",false);
      displayObject(rectangle);

      GeometricObject circle = new Circle(14,"Yellow",true);
      displayObject(circle);
   }

   public static void displayObject(Object object){
      System.out.print(object);
   }
}



import java.util.Date;

abstract class GeometricObject{
        private String color;
        private boolean filled;
        private Date dateCreated;

        protected GeometricObject(){
                this.color = "Black";
                this.filled = false;
                this.dateCreated = new Date();
        }

        protected GeometricObject(String color,boolean filled){
                this.color = color;
                this.filled = filled;
                this.dateCreated = new Date();
        }
```

```
	//setter

	public void setColor(String color){
		this.color = color;
	}

	public void setFilled(){
		this.filled = filled;
	}


	//getter

	public String getColor(){
		return color;
	}

	public boolean isFilled(){
		return filled;
	}

	public Date getDate(){
		return dateCreated;
	}

	//abstract method
	abstract public double getArea();
	abstract public double getPerimeter();

	@Override
	public String toString(){
		return "\nColor: "+color+"\nFilled: "+filled +"\n";
	}
}



class Rectangle extends GeometricObject{
	private double sideA,sideB;
	private double perimeter,area;

	public Rectangle(){
		super("Black",false);
		this.sideA = 1;
```

```
            this.sideB = 1;
        }

        public Rectangle(double sideA,double sideB,String color,boolean filled){
            super(color,filled);
            this.sideA = sideA;
            this.sideB = sideB;
        }

        @Override
        public double getArea(){
    return sideA*sideB;
        }

        @Override
        public double getPerimeter(){
            return (sideA + sideB)*2;
        }

        public String toString(){
            return "Rectangle: side1 = " + sideA +
            " side2 = " + sideB + "\nPerimeter: " + String.format("%.2f",getPerimeter())+
            "\nArea: "+String.format("%.2f",getArea()) +
            super.toString();
        }
}



class Triangle extends GeometricObject{
        private double sideA,sideB,sideC;
        private double perimeter,area;

        public Triangle(){
            this.sideA = 1;
            this.sideB = 1;
            this.sideC = 1;
        }

        public Triangle(double sideA,double sideB,double sideC,String color,boolean filled){
            super(color,filled);
            this.sideA = sideA;
            this.sideB = sideB;
            this.sideC = sideC;
```

```
	}

	@Override
	public double getArea(){
		double s = (sideA + sideB + sideC) / 2;
      return Math.sqrt(s * (s - sideA) * (s - sideB) * (s - sideC));
	}

	@Override
	public double getPerimeter(){
		return sideA + sideB + sideC;
	}

	public String toString(){
		return "Triangle: side1 = " + sideA +
		" side2 = " + sideB + " side3 = " + sideC + "\nPerimeter: " +
String.format("%.2f",getPerimeter())+
		"\nArea: "+String.format("%.2f",getArea()) +
		super.toString();
	}
}



class Circle extends GeometricObject{
	private double diameter;
	private double perimeter,area;

	public Circle(){
		super("Black",false);
		this.diameter = 1;
	}

	public Circle(double diameter,String color,boolean filled){
		super(color,filled);
		this.diameter = diameter;
	}

	@Override
	public double getArea(){
      return Math.PI*(diameter/2)*(diameter/2);
	}

	@Override
```

```
	public double getPerimeter(){
		return 2*Math.PI*(diameter/2);
	}

	public String toString(){
		return "\nCircle: diameter = " + diameter +
		"\nPerimeter: " + String.format("%.2f",getPerimeter())+
		"\nArea: "+String.format("%.2f",getArea()) +
		super.toString();
	}
}
```