```
+----------------------+
| [1] TRIANGLE ABSTRACT |
+----------------------+
package Task1;

import java.util.Date;
public abstract class GeometricObject {

        private String color;
        private boolean filled;
        private java.util.Date dateCreated;

        protected GeometricObject() {
                super();
                dateCreated = new java.util.Date();
        }
        protected GeometricObject(String color, boolean filled) {
                super();
                this.color = color;
                this.filled = filled;
                dateCreated = new java.util.Date();
        }
        public String getColor() {
                return color;
        }
        public void setColor(String color) {
                this.color = color;
        }
        public boolean isFilled() {
                return filled;
        }
        public void setFilled(boolean filled) {
                this.filled = filled;
        }
        public java.util.Date getDateCreated() {
                return dateCreated;
        }
        @Override
        public String toString() {
                return " [color = " +color+ ", filled = " +filled+ ", Created on = "
+dateCreated+ "]";
        }
        public abstract double getArea();
        public abstract double getPerimeter();

}
package Task1;
public class Triangle extends GeometricObject {

        private double side1 , side2 , side3;
```

```
	protected Triangle() {
		super();
		side1 = side2 = side3 = 1.0 ;
	}
	protected Triangle(double side1, double side2, double side3) {
		super();
		this.side1 = side1;
		this.side2 = side2;
		this.side3 = side3;
	}
	protected Triangle(String color, boolean filled, double side1, double side2,
double side3) {
		super(color, filled);
		this.side1 = side1;
		this.side2 = side2;
		this.side3 = side3;
	}

	public double getSide1() {
		return side1;
	}
	public void setSide1(double side1) {
		this.side1 = side1;
	}
	public double getSide2() {
		return side2;
	}
	public void setSide2(double side2) {
		this.side2 = side2;
	}
	public double getSide3() {
		return side3;
	}
	public void setSide3(double side3) {
		this.side3 = side3;
	}
	public double getArea() {
		double s = side1 + side2 + side3;
		return Math.sqrt(s * (s - side1) * (s - side2) * (s - side3));
	}
	public double getPerimeter() {
		return side1 + side2 + side3;
	}

}
package Task1;
import java.util.Scanner;

public class TestTriangle {
```

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

                Scanner input = new Scanner(System.in);
                Triangle tri = new Triangle();

                System.out.println("Enter three sides of Triangle : ");
                tri.setSide1(input.nextDouble());
                tri.setSide2(input.nextDouble());
                tri.setSide3(input.nextDouble());

                input.nextLine();
                System.out.println("Enter color : ");
                tri.setColor(input.nextLine());

                System.out.println("IsFilled : ");
                tri.setFilled(input.nextBoolean());

                System.out.println("Area : " +tri.getArea()+ "\nPerimeter : "
+tri.getPerimeter()+ "\nColor : " +tri.getColor()+ "\nIsFilled? " +tri.isFilled());

                System.out.println("\n\n" +tri.toString());

        }

}
================================================================================
================================================================================
+--------------+
| [2] CALENDAR |
+--------------+
package Task2;
import java.util.Scanner;

public class PrintCalendar {
  public static void main(String[] args) {

    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter full year [Example :  2019] : ");
    int year = scanner.nextInt();

    System.out.print("Enter month in number between 1 and 12: ");
    int month = scanner.nextInt();

    printMonth(year, month);

  }

public static void printMonth(int year, int month) {
```

```
    printMonthTitle(year, month);
    printMonthBody(year, month);

  }
public static void printMonthTitle(int year, int month) {

    System.out.println("         " + getMonthName(month)+ " " + year);
    System.out.println("-----------------------------");
    System.out.println(" Sun Mon Tue Wed Thu Fri Sat");

  }
public static String getMonthName(int month) {

    String monthName;

    switch (month) {
    case 1: monthName = "January"; break;
    case 2: monthName = "February"; break;
    case 3: monthName = "March"; break;
    case 4: monthName = "April"; break;
    case 5: monthName = "May"; break;
    case 6: monthName = "June"; break;
    case 7: monthName = "July"; break;
    case 8: monthName = "August"; break;
    case 9: monthName = "September"; break;
    case 10: monthName = "October"; break;
    case 11: monthName = "November"; break;
    case 12: monthName = "December"; break;
    default : monthName = "Wrong input";
  }
  return monthName;

}

public static void printMonthBody(int year, int month) {

  int startDay = getStartDay(year, month);
  int numberOfDaysInMonth = getNumberOfDaysInMonth(year, month);

  for (int i = 0; i < startDay; i++) {
        System.out.print("    ");
  }



  for (int i = 1; i <= numberOfDaysInMonth; i++) {
    if (i < 10)
      System.out.print("   " + i);
   else
      System.out.print("  " + i);
```

```
    if ((i + startDay) % 7 == 0)
      System.out.println();
  }
  System.out.println();
}

public static int getStartDay(int year, int month) {

  // Get total number of days since 1/1/1800
  int startDay1800 = 3;
  int totalNumberOfDays = getTotalNumberOfDays(year, month);
  return (totalNumberOfDays + startDay1800) % 7;
}

// Get the total number of days since January 1, 1800
public static int getTotalNumberOfDays(int year, int month) {

  int total = 0;
  // Get the total days from 1800 to year - 1
  for (int i = 1800; i < year; i++) {
  if (isLeapYear(i))
    total = total + 366;
  else
    total = total + 365;
  }

  // Add days from Jan to the month prior to the calendar month
  for (int i = 1; i < month; i++) {
    total = total + getNumberOfDaysInMonth(year, i);
  }

  return total;
}
//Get the number of days in a month
public static int getNumberOfDaysInMonth(int year, int month) {

  if (month == 1 || month == 3 || month == 5 || month == 7 || month == 8 || month ==
10 || month == 12)
    return 31;
  if (month == 4 || month == 6 || month == 9 || month == 11)
    return 30;
  if (month == 2) return isLeapYear(year) ? 29 : 28;

  return 0; // If month is incorrect
}
// Determine if it is a leap year

public static boolean isLeapYear(int year) {
  return year % 400 == 0 || (year % 4 == 0 && year % 100 != 0);
```

```
  }

}
================================================================================
================================================================================
+-----------------------+
| [3] COLORABLE SQUARE  |
+-----------------------+ [[ USING ABSTRACT GEOMTERICOBJECT CLASS ]]
package Task3;

public interface Colorable {
        public abstract String howToColor();

}
package Task3;

public class Square extends GeometricObject implements Colorable{
        private double side;

        public Square() {
                super();
                side = 1.0;
        }
        public Square(double side) {
                super();
                this.side = side;
        }
        public Square(String color, boolean filled, double side) {
                super(color, filled);
                this.side = side;
        }
        public double getSide() {
                return side;
        }
        public void setSide(double side) {
                this.side = side;
        }
        @Override
        public String toString() {
                return "Square : " +super.toString()+ " side = " +side+ "]";
        }
        @Override
        public double getArea() {
                return side * side;
        }
        @Override
        public double getPerimeter() {
                return side * 4;
        }
```

```
        @Override
        public String howToColor() {
                return " Color all four sides.";
        }

}
package Task3;
import java.util.Scanner;

public class Test {
        public static void main(String [] args) {
                Scanner input = new Scanner (System.in);
                GeometricObject [] squareList = new GeometricObject [5];

                for(int i = 0 ;  i < 5 ; i++) {
                        System.out.println("Enter square side : ");
                        squareList[i] = new Square (input.nextInt());
                        Square sq = (Square)squareList[i];

                        System.out.println("Area = " +sq.getArea());
                        System.out.println(sq.howToColor());

                }
        }

}
```

==========

==========

```
+----------------------+
| [4] COMPARABLE CIRCLE |
+----------------------+ [[ USING ABSTRACT GEOMETRICOBJECT CLASS & CIRCLE CLASS ]]

package Task3;

public class Circle extends GeometricObject implements Comparable<Circle> {
        private double radius;

        public Circle() {
                super();
                radius = 1.0;
        }
        public Circle(double radius) {
                super();
                this.radius = radius;
        }
        public Circle(String color, boolean filled, double radius) {
                super(color, filled);
                this.radius = radius;
        }
        public double getRadius() {
```

```
                return radius;
        }
        public void setRadius(double radius) {
                this.radius = radius;
        }
        @Override
        public String toString() {
                return "Circle : " +super.toString()+ " radius=" +radius+ "]";
        }
        @Override
        public double getArea() {
                return Math.PI * radius * radius;
        }
        @Override
        public double getPerimeter() {
                return Math.PI * 2 * radius;
        }
        @Override
        public int compareTo(Circle o) {
                if(this.radius > o.getRadius())
                        return 1 ;
                else if(this.radius < o.getRadius())
                        return -1 ;
                else
                        return 0;
        }
        @Override
        public boolean equals(Object obj) {
                return this.compareTo((Circle)obj) == 0;
        }

}
package Task3;

public class TestComparableCircle {
        public static void main(String[] args) {

                Circle c1 = new Circle(5);
                Circle c2 = new Circle(5);
                Circle c3 = new Circle(10);

                System.out.println("Circle1 equals to Circle2 ? " +c1.equals(c2));
                System.out.println("Circle1 equals to Circle3 ? " +c1.equals(c3));
                System.out.println("Circle2 equals to Circle3 ? " +c2.equals(c3));
        }

}
================================================================================
================================================================================
+---------------------------------+
```

| [5] AREA OF ALL GEOMETRIC OBJETCS |
+----------------------------------+ [[USING ABSTRACT GEOMETRICOBJECT CLASS]]

```
package Task5;

public abstract class GeometricObject {
	private String color;
	private boolean filled;
	private java.util.Date dateCreated;

	protected GeometricObject() {
		super();
		color = " white ";
		filled = false;
		dateCreated = new java.util.Date();
	}
	protected GeometricObject(String color, boolean filled) {
		super();
		this.color = color;
		this.filled = filled;
		dateCreated = new java.util.Date();
	}
	public String getColor() {
		return color;
	}
	public void setColor(String color) {
		this.color = color;
	}
	public boolean isFilled() {
		return filled;
	}
	public void setFilled(boolean filled) {
		this.filled = filled;
	}
	public java.util.Date getDateCreated() {
		return dateCreated;
	}
	@Override
	public String toString() {
		return " [color = " +color+ ", filled = " +filled+ ", Created on = "
+dateCreated.toString();
	}
	public abstract double getArea();

	public abstract double getPerimeter();

}
package Task5;

public class Circle extends GeometricObject implements Comparable <Circle> {
	private double radius;
```

```
	public Circle() {
		super();
		radius = 1.0;
	}
	public Circle(double radius) {
		super();
		this.radius = radius;
	}
	public Circle(String color, boolean filled, double radius) {
		super(color, filled);
		this.radius = radius;
	}
	public double getRadius() {
		return radius;
	}
	public void setRadius(double radius) {
		this.radius = radius;
	}
	@Override
	public String toString() {
		return "Circle : " +super.toString()+ " radius = " +radius+ "]";
	}
	@Override
	public int compareTo(Circle o) {
		if(this.radius > o.getRadius())
			return 1;
		else if(this.radius < o.getRadius())
			return -1;
		else
			return 0;
	}
	@Override
	public double getArea() {
		return Math.PI * radius * radius;
	}
	@Override
	public double getPerimeter() {
		return Math.PI * radius * 2;
	}
	@Override
	public boolean equals(Object obj) {
		return this.compareTo((Circle)obj) == 0;
	}

}
package Task5;

public class Rectangle extends GeometricObject implements Comparable <Rectangle> {
	private double height , width;
```

```
	public Rectangle() {
		super();
		height = 1;
		width = 1;
	}
	public Rectangle(double height, double width) {
		super();
		this.height = height;
		this.width = width;
	}
	public Rectangle(String color, boolean filled, double height, double width)
{
		super(color, filled);
		this.height = height;
		this.width = width;
	}
	public double getHeight() {
		return height;
	}
	public void setHeight(double height) {
		this.height = height;
	}
	public double getWidth() {
		return width;
	}
	public void setWidth(double width) {
		this.width = width;
	}
	@Override
	public double getArea() {
		return height * width;
	}
	@Override
	public double getPerimeter() {
		return 2 * (height + width);
	}
	@Override
	public int compareTo(Rectangle o) {
		if(this.getArea() > o.getArea())
			return 1;
		else if(this.getArea() < o.getArea())
			return -1;
		else
			return 0;
	}
	@Override
	public boolean equals(Object obj) {
		return this.compareTo((Rectangle)obj) == 0;
	}
```

```
        @Override
        public String toString() {
                return "Rectangle : " +super.toString()+ " height = " +height+ ",
width = " +width+ "]";
        }

}
package Task5;

public class Test {
        public static void main(String[] args) {

                GeometricObject [] list = new GeometricObject[4];

                list[0] = new Circle(5);
                list[1] = new Circle(2);
                list[2] = new Rectangle(5, 3);
                list[3] = new Rectangle(1, 2);

                System.out.println("Total area : " +sumArea(list));

        }
        public static double sumArea(GeometricObject[] a) {
                double sum = 0;
                for(int i = 0 ; i < a.length ; i++) {
                        sum+= a[i].getArea();
                }
                return sum;
        }

}
```

==========

==========