Assignment A2 (35 marks)

Focus: *Abstract Classes and Interfaces, Standard Java Interfaces: Comparable, Cloneable,*

For this assignment, you need to download from Canvas the Shape class shown below and include it in your project.

```
public abstract class Shape {
    //attributes
    private String color;
    private boolean filled;
    //constructors
    protected Shape(){this("White", true);}
    protected Shape(String color, boolean filled){
        setColor(color);
        setFilled(filled);
    }
    //methods
    public String getColor(){return color;}
    public void setColor(String color){this.color = color;}
    public boolean isFilled(){return filled;}
    public void setFilled(boolean f){filled = f;}
    public String toString(){
        return "Color: " + color + ". " + (filled? "Filled. ":"Not filled. ");
    }
    public abstract double getArea();
    public abstract double getPerimeter();
}
```

Q1. [7 marks] Modify the class Shape so that it implements Cloneable and Comparable interfaces. The comparison should be solely based on the area of two shapes. Use the following method header: **public int compareTo(Shape shp)**

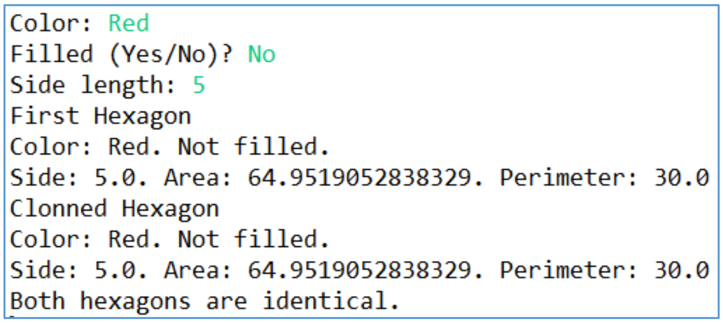
Q2. [11 marks] Create a class named Hexagon that extends Shape.

Assume that all six sides of a hexagon are of equal length. Create appropriate setter and getter method(s) and constructor(s), and override the toString method to return a string representation of all attribute values as well as the area and perimeter of a hexagon object.

Write a test program that asks the use for the attributes of a Hexagon object (i.e., side length, color, and filled), creates that object, and then displays its state (side, color, area, etc.). Assume users enter valid inputs. Create a new object using the clone method and compare the two objects using the compareTo method.

Hint: Add “throws CloneNotSupportedException” to the header of the clone method as well as the main method in your test program. You will learn more about exception handling in next lecture and lab.

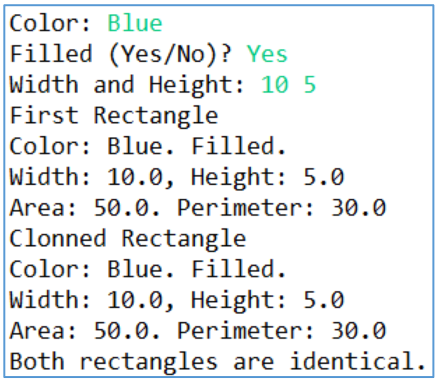
Sample run



```
Color: Red
Filled (Yes/No)? No
Side length: 5
First Hexagon
Color: Red. Not filled.
Side: 5.0. Area: 64.9519052838329. Perimeter: 30.0
Clonned Hexagon
Color: Red. Not filled.
Side: 5.0. Area: 64.9519052838329. Perimeter: 30.0
Both hexagons are identical.
```

Q3. [7 Marks] Repeat Q1 for a `Rectangle` class (note: width is not always equal to height).

Sample run

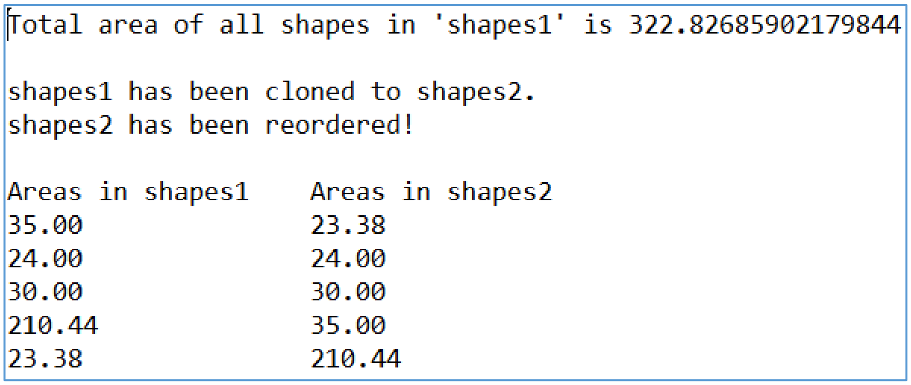


```
Color: Blue
Filled (Yes/No)? Yes
Width and Height: 10 5
First Rectangle
Color: Blue. Filled.
Width: 10.0, Height: 5.0
Area: 50.0. Perimeter: 30.0
Clonned Rectangle
Color: Blue. Filled.
Width: 10.0, Height: 5.0
Area: 50.0. Perimeter: 30.0
Both rectangles are identical.
```

Q4. [10 marks] Create a new test program with a main method that does the following:

- Creates an array named `shapes1` that has five objects: three of the type `Rectangle` and two of the type `Hexagon`. Assume any values for the object attributes.
- Calculates and prints the total area of all objects in `shapes1`.
- Creates a clone of `shapes1`. Name your clone `shapes2`.
- Sorts the elements in `shapes2`. *Hint: use a method from `java.util.Arrays` class.*
- Displays the areas of all shapes in both arrays. *Hint: use `printf()` to have your output formatted similarly to the sample run below.*

Sample run



```
Total area of all shapes in 'shapes1' is 322.82685902179844

shapes1 has been cloned to shapes2.
shapes2 has been reordered!

Areas in shapes1    Areas in shapes2
35.00               23.38
24.00               24.00
30.00               30.00
210.44              35.00
23.38               210.44
```

Grading:

Q1: +2 class header
+3 compareTo
+2 clone

Q2: +1 for class header
+2 for side, constructor, setter and getter methods
+2 for getArea and getPerimeter methods
+2 for toString
+4 for Test program

Q3: +1 for class header
+2 for attributes, constructor, setter, getter, getArea, getPerimeter methods
+2 for toString
+2 for Test program

Q4: +3 for shapes1 declaration and initialization
+3 for area calculation
+2 for cloning
+2 for displaying areas in both arrays

Submission Instructions

For programming questions, explain in few, simple sentences **the algorithm you used to tackle the problem**. Add these sentences as a **block comment at the beginning of your program**. For coding questions, make sure to use appropriate code formatting and structure (e.g., indentation, brackets, etc.).

For this assignment, you need to do the following:

1- Create a Java project of which name consists of **your student number followed by the assignment number,** e.g., "1234567_A2".
2- Create one class for each question and write your answer inside that class. Your classes should have the same name as the question number (e.g., Q1)
3- After solving all questions, open Windows Explorer (or any other file explorer).
4- Navigate to your Java project folder (can be found inside your Eclipse workspace folder).
5- Locate the "src" folder for this project (the folder that includes the source code for all questions).
6- Zip the "src" folder and rename the zipped file to match your project name (e.g., 1234567_A2.zip).
7- Submit the zipped file **to Canvas.**

Note that you can resubmit an assignment, but the new submission overwrites the old submission and receives a new timestamp.