CS 9053 – Section 2
Wednesday, March 25, 2020
Dr. Dean Christakos

Assignment 5
Due: March 31, 2020

Part I:

Problem Description:

(*Enabling GeometricObject comparable*) Modify the GeometricObject class to implement the Comparable interface, and define a static max method in the GeometricObject class for finding the larger of two GeometricObject objects.

Write a test program that uses the max method to find the larger of two circles and the larger of two rectangles.

```
public class Test {
  // Main method
  public static void main(String[] args) {
    // Create two comparable circles
    Circle1 circle1 = new Circle1(5);
    Circle1 circle2 = new Circle1(4);

    // Display the max circle
    Circle1 circle = (Circle1)GeometricObject1.max(circle1, circle2);
    System.out.println("The max circle's radius is " +
      circle.getRadius());
    System.out.println(circle);
  }
}

abstract class GeometricObject1 implements Comparable {
  // Implement it
}

// Circle.java: The circle class that extends GeometricObject
class Circle1 extends GeometricObject1 {
  // Implement it
}
```

Part II:

Problem Description:

A complex number is a number of the form $a+bi$, where a and b are real numbers and i is $\sqrt{-1}$. The numbers a and b are known as the real part and imaginary part of the complex number, respectively. You can perform addition, subtraction, multiplication, and division for complex numbers using the following formula:

$$a+bi+c+di=(a+c)+(b+d)i$$
$$a+bi-(c+di)=(a-c)+(b-d)i$$
$$(a+bi)*(c+di)=(ac-bd)+(bc+ad)i$$
$$(a+bi)/(c+di)=(ac+bd)/(c^2+d^2)+(bc-ad)i/(c^2+d^2)$$

You can also obtain the absolute value for a complex number using the following formula:

$$|a+bi|=\sqrt{a^2+b^2}$$

(A complex number can be interpreted as a point on a plane by identifying the (a,b) values as the coordinates of the point. The absolute value of the complex number corresponds to the distance of the point to the origin, as shown in Figure 13.12b.)

Design a class named Complex for representing complex numbers and the methods add, subtract, multiply, divide, abs for performing complex-number operations, and override toString method for returning a string representation for a complex number. The toString method returns a + bi as a string. If b is 0, it simply returns a.

Provide three constructors Complex(a, b), Complex(a), and Complex(). Complex() creates a Complex object for number 0 and Complex(a) creates a Complex object with 0 for b. Also provide the getRealPart() and getImaginaryPart() methods for returning the real and imaginary part of the complex number, respectively.

Your Complex class should also implement the Cloneable interface.

Write a test program that prompts the user to enter two complex numbers and display the result of their addition, subtraction, multiplication, and division. Here is a sample run:

<Output>
```
Enter the first complex number: 3.5 5.5
Enter the second complex number: -3.5 1
(3.5 + 5.5i) + (-3.5 + 1.0i) = 0.0 + 6.5i
(3.5 + 5.5i) - (-3.5 + 1.0i) = 7.0 + 4.5i
(3.5 + 5.5i) * (-3.5 + 1.0i) = -17.75 + -15.75i
(3.5 + 5.5i) / (-3.5 + 1.0i) = -0.5094 + -1.7i
|3.5 + 5.5i| = 6.519202405202649
```

<End Output>

The template for the code is:

```java
import java.util.Scanner;

public class Test {
  public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    System.out.print("Enter the first complex number: ");
    double a = input.nextDouble();
    double b = input.nextDouble();
    Complex c1 = new Complex(a, b);

    System.out.print("Enter the second complex number: ");
    double c = input.nextDouble();
    double d = input.nextDouble();
    Complex c2 = new Complex(c, d);

    System.out.println("(" + c1 + ")" + " + " + "(" + c2 + ")" + " = " +
c1.add(c2));
    System.out.println("(" + c1 + ")" + " - " + "(" + c2 + ")" + " = " +
c1.subtract(c2));
    System.out.println("(" + c1 + ")" + " * " + "(" + c2 + ")" + " = " +
c1.multiply(c2));
    System.out.println("(" + c1 + ")" + " / " + "(" + c2 + ")" + " = " +
c1.divide(c2));
    System.out.println("|" + c1 + "| = " + c1.abs());

    Complex c3 = (Complex)c1.clone();
    System.out.println(c1 == c3);
    System.out.println(c3.getRealPart());
    System.out.println(c3.getImaginaryPart());
  }
}

class Complex {
  // Write your code
}
```

Part III

This is a followup to Part II.

A complex number is a type of Number. You should have figured that out, so I don’t feel like I am giving anything away.

Create a class called **maxFinder** with a Generic constraint into which you can add a collection of Numbers and has a method max() which will return the largest value within that collection. How you implement it is up to you, but it should have methods:

add(T t) -> add an object of type T
T max() -> return the maximum valued object, with return type T

The best implementation which will get full credit will be a class that only accepts Comparable objects when creating it using generics.