CSC 115
Spring 2014

Assignment #1

Due

Before 8:30 pm on Friday January 24, 2014. You can only consider you assignment submitted when you have received a message from Connex system that indicates your assignment has been submitted. Save that message.

Description – Part 1

Learning Outcomes: Upon successful completion of Assignment #1 Part 1 you will be able to:

- Use a Java class to instantiate objects and manipulate its attributes using the class methods.
- Write Java programs that are able to input data from a text file.

Your Tasks:

1. Carefully review the **`Complex1.java`** class provided. In part 1 (this part) of Assignment 1 you are not expected to alter the coding in this file in any way. However, you need to add the following documentation to **`Complex1.java`**:
   - Immediately prior to each of the *constructor* methods add (appropriately) one of the following comments:

     ```
     // Default Constructor

     // Constructor for a new complex number, with:
     //   - real component = r
     //   - imaginary component = i
     ```

   - Immediately prior to each of the *accessor* methods add a comment with the format below, replacing the `XXX` with the appropriate attribute name:

     ```
     // Accessor for the XXX attribute
     ```

   - Immediately prior to each of the *mutator* methods add a comment with the format below, replacing the `XXX` with the appropriate attribute name:

     ```
     // Mutator for the XXX attribute
     ```

   - Test that the added documentation did not affect the code by compiling:

     ```
     javac Complex1.java
     ```

     The result should be a simple success message.

2. Using the **`Complex1.java`** class provided write your own Java program, called **`ComplexExerciser1.java`**, that instantiates complex numbers (ie, calls the constructor of the Complex class) and then outputs them, as follows:

   - one with value `2 + 4i`,
   - another with value `4 + -5i`, and
   - an array of complex numbers that contain the values that are found in the file `ComplexData.txt`. The file will contain a single integer on the first line indicating the number of lines that follow, then each of the following lines will contain two integers, the first one will be used for the real part of a complex number, the second one will be used for the imaginary part of the same complex number. (An example input file is provided.)
   - After all of the numbers have been created, print them to the output screen, using calls to the `toString()` method of the `Complex1` class.
   - Add documentation to the file includes a name, purpose, author, edit date, list of credits.

The solution to the problem (stated above) must be contained in one Java file, called **`ComplexExerciser1.java`**.

The **`Complex1.java`** class must be used, including using its methods, wherever possible but none of the attributes nor method code should altered in any way.

Sample Output:

Test that your program functions correctly by (first) compiling it:

```
javac ComplexExerciser1.java
```

Once it is compiled without errors, execute the program:

```
java ComplexExerciser1
```

An output using the example data as provided in the `ComplexData.txt` file is as follows:

```
First complex number:2 + 4i
Second complex number:4 + -5i
From File:
2 + -2i
132 + 0i
-17 + 6i
2 + 1i
4 + -1i
0 + 7i
```

Because this assignment will be graded using an automated marking system your output, using this example input, should appear ***exactly*** the same, including spacing and line wraps, as the above.

Submitting your Solution:

When complete (including suitable documentation) submit your **`Complex1.java`** and **`ComplexExerciser1.java`** files to the CSc 115 Connex Site using the Assignments: Assignment 1. Due: Before 8:30 pm on Friday January 24, 2014.

Before you go on: and in case you don't remember complex numbers from your previous math courses: A description of the theory and some exercises are provided in the file Complex Numbers Worksheet. Please complete the exercises on the worksheet.

Description - Part 2

Learning Outcomes: Upon successful completion of Assignment #1 Part 2 you will be able to:

- Discern the difference between *static* methods and *instance* methods
- Add instance methods to a Java class.
- Test the functionality of the created class by instantiating objects and manipulating attributes using instance methods.

Your Tasks:

The **`Complex2.java`** file provided is the same as Complex1.java with the addition of a tester (a main method used to test instance methods within the class) and 'stub' methods that will be expanded by you. Use **`Complex2.java`** for this part of the assignment.

- Adjust the **`toString()`** method in **`Complex2.java`** such that it improves the appearance of complex numbers with imaginary parts that are `0` or negative, as follows:
  - When the imaginary component of a Complex2 object is negative, replace the plus sign (+) between the real and the imaginary parts of the number with a minus sign (-).
  - When the imaginary component of the Complex2 object is zero, do not output the imaginary part or the sign between the parts.

  For example:

Correct output: `3 - 2 i`	Incorrect output: `3 + -2 i`
Correct output: `7`	Incorrect output: `7 + 0 i`

- Create another constructor method in **`Complex2.java`**: It will have only one integer parameter that will be used for the real attribute; the corresponding imaginary attribute will be equal to 0.
- Write method called `add` with the following method signature:

  ```
  public Complex2 add(Complex2 val)
  ```

  The method you write need will return a complex number whose value is the sum of the existing (Complex2) number plus the parameter `val`.

- Write a method called subtract with the following method signature:
  `public Complex2 subtract(Complex2 val)`
  The method you write need will return a complex number whose value is the difference between the existing (Complex2) number minus the parameter `val`.
- Write method called multiply with the following method signature:
  `public Complex2 multiply(Complex2 val)`
  The method you write need will return a complex number whose value is the product of the existing (Complex2) number multiplied by the parameter `val`.
- Write method called divide with the following method signature:
  `public Complex2 divide(Complex2 val)`
  The method you write need will return a complex number whose value is the quotient of the existing (Complex2) divided by the parameter `val`. (The resulting complex number *does* have integer valued attributes, thus truncation of fractional results should occur.)
- Add a comment at the top of your Complex2.java file that explains the difference between a *static* method (example: main) and the instance methods (example: add, subtract, multiply, divide, . . .) .
- Add documentation to your Complex2.java file. Suitable documentation for this assignment is:
  - ✓ The file needs a name, purpose, author, edit date, list of credits
  - ✓ Every method needs a description, input, output
  - ✓ Throughout the code, every 3-6 lines, that form a functional group, require a descriptive comment.

Test the functionality of each new method using the Tester method provided in Complex2.java. Compile it using:

```
javac Complex2.java
```

Once it is compiled without errors, execute the tester using:

```
java Complex2
```

The output of your tester should be the following:

```
Complex Number Tester Output:
Real Constructor Test: Should Output 423 :  423
Add Tester: Should Output -1 + 3i :  -1 + 3i
Subtract Tester: Should Output -5 + 5i :  -5 + 5i
Multiply Tester: Should Output -2 + 11i :  -2 + 11i
Divide Tester: Should Output -2 + 1i :  -2 + 1i
```

Submitting your Solution:

- When complete (including suitable documentation) submit your files to the CSc 115 Connex Site using the Assignments: Assignment 1 link.

Due: Before 8:30 pm on Friday January 24, 2014.

Observe: Your *complete* Assignment 1 submission requires 3 java files: `Complex1.java`, `ComplexExerciser1.java` and `Complex2.java`.