Quiz 4.b (Take Home)

Q1. Define a class for complex numbers.

A complex number is a number of the form *a + b*i* where, for our purposes, a and b are numbers of type **`double`**, and i is a number that represents the quantity $\sqrt{-1}$.

Part One: Represent a complex number as two values of type **`double`**. Name the instance variables **`real`** and **`imaginary`**. (The instance variable for the number that is multiplied by i is the one called imaginary.) Call the class **`Complex`**. Include a constructor with two parameters of type **`double`** that can be used to set the instance variables of an object to any values. Also include a constructor that has only a single parameter of type **`double`**; call this parameter **`realPart`** and define the constructor so that the object will be initialized to **`realPart+0*i.`** Also include a no-argument constructor that initializes an object to 0 (that is, to **`0+0*i`**). Define accessor and mutator methods as well as the methods **`equals`** and **`toString`**. Define static methods for addition, subtraction, and multiplication of objects of your class **`Complex`**. These methods should be static and should each have two parameters of type **`Complex`** and return a value of type **`Complex`**. For example, **`Complex.add(c1, c2)`** will return the result of adding the two complex numbers (two objects of Complex class) c1 and c2.

> *Hints:* To add or subtract two complex numbers, you add or subtract the two instance variables of type double. The product of two complex numbers is given by the following formula:
>
> $$(a + b*i)*(c + d*i) = (a*c - b*d) + (a*d + b*c)*i$$

Part Two: Add a second version of the methods for addition, subtraction, and multiplication. These methods should have the same names as the static version but should use a calling object and a single argument. For example, this version of the add method (for addition) has a calling object and one argument. So **`c1.add(c2)`** returns the result of adding the complex numbers c1 and c2. Note that your class should have all these methods; for example, there should be two methods named add.

Part Three: Add another version of the methods for addition, subtraction, and multiplication. These methods should not have the same names as the static version

(do you know why?) and should use a calling object and a single argument. The methods will be **`void`** methods. . Method names should be plus, minus, and times. The result is given as the changed value of the calling object. For example, this version of the add method (for addition) has a calling object and one argument. Therefore, **`c1.plus(c2);`** changes the values of the instance variables of c1 so they represent the result of adding c2 to the original version of c1.

Q2. Write a Calculator User Interface as Follows:

The program should be a menu driven program that works infinitely or tell a user requests quitting. Main menu should list the possible types of numbers and the quitting option (i.e. 1.Real, 2.Complex, 3.Quit) Under each of the first two options a submenu containing the operations should be provided. The program should be fully implemented having a clear dialogue with the user and providing results in a nice format (i.e. 4 + 3 = 7, not only 7).

DELIVERY GUIDELINES:

1. Fully working softcopy version should be submitted in WebCT.
2. Report containing the problem definition, screen snapshots, and how valuable you found the Quiz in learning Java.
3. A test class must be provided to *each* Part of Q1.
4. Plagiarism will lead to F in the whole lab grade.
5. You have to drop by my office to discuss your working code.
6. [further modifications might be announced, keep in touch!]

~ Good Luck! ~