

1. Create a class called **Complex** for performing arithmetic with complex numbers. Complex numbers have the form: realPart + imaginaryPart * i

So, This class has two data members: realPart, imaginaryPart (use double to represent your data members)

Provide also the following member functions:

- A constructor with no arguments that initializes realPart and imaginaryPart to zero.
- An overloaded constructor with arguments.
- The “Norm” function that computes and returns the norm of a complex number:

The norm of realPart + imaginaryPart * i is
 $\text{sqrt}(\text{realPart}^2 + \text{imaginaryPart}^2)$

- A “Print” function that prints on the screen a complex number as follows:
If realPart=2 and imaginaryPart=3 it should print: **(2 + 3i)**
- An “add” function that adds two complex numbers: the real parts are added together and the imaginary parts are added together. It will be used as follows:

```
Complex c(3,6);  
Complex d(1,3);  
Complex e;  
e.add(c,d); // e should become (4,9) after this statement
```

Write a client program (main function) which creates objects of class Complex and test all your member functions.

2. Create a class **advisor** with the following data members:

- **id** The advisor's ID.
An integer number representing the advisor's identification number. The valid values are between 1000 and 9000 inclusive. (Defaults to **0**)
- **name** The advisor's full name. Ex: James Bond
String of 40 characters. (Defaults to **Empty String**)
- **rank** The advisor's rank
Valid values are 'A' for regular Advisor, 'C' for Chairperson and 'D' for Dean. (Defaults to "A")
- **password** The advisor's password to login to the Student Information System.
A string that has exactly 8 characters (not less, not more). It must be a combination of letters and digits. It cannot contain spaces. (Defaults to **Empty String**)
- **nbrAdvisees** The number of advisees
An integer number representing the number of advisees. The valid values are between 1 and 50 inclusive. (Defaults to **15**)

And the following member functions:

- A constructor **with default arguments**
- All "set" functions with the **integrity checks**
- All "get" functions.
- A member function that increments the **nbrAdvisees** by a certain value. Note that this function should verify that **nbrAdvisees** is still valid.
- A **print** function to print all the attributes of an advisor: id, name, rank, password and nbrAdvisees. This function should also generate and print the email of the advisor. You should generate the email as follows: the first letter of the first name, followed by the last name, followed by "@ndu.edu.lb". For example, the email of James Bond would be JBond@ndu.edu.lb

Write a program to test your class by:

- Create an NDU advisor list of 3 advisors (array of 3 objects)
- Fill the objects of your array with advisor's data of your choice.
- Using a pointer, print on the screen the created advisors' objects
- Test your member functions using objects of the created array.

Submission

The following should be returned to the instructor in a folder:

1. A first page with your name, the assignment number.
2. Problem Description.
3. A listing of your programs (class and client program) **well documented**.
4. A screen capture of the output of your program.

Due date: Monday, October 22, 2012