CMP 210 – Data Structures and Algorithms
Spring 2021
LAB-1

Issue Date: February 26, 2021

The objective of this lab is to:

1. Refresh OOP concepts focusing on Abstract Data Types, dynamic allocation and array of objects.
2. Reading and following coding conventions.
3. Debugging.

ALERT!

1. Bringing your laptops is a **MUST** to attend the lab, if you do not have your laptop with you leave the lab silently and enjoy the cafeteria.
2. This is an individual lab, you are strictly **NOT** allowed to discuss your solution with fellow colleagues, even not allowed asking how is he/she is doing, it may result in negative marking. You can **ONLY** discuss with your TAs or with me.
4. Strictly follow good coding conventions (commenting, meaningful variable and functions names, properly indented and modular code.
5. Save your work frequently. Make a habit of pressing **CTRL+S** after every line of code you write.
3. **Anyone caught in act of plagiarism would be awarded an "F" grade in this Lab.**

Task 01: [15 Minutes]

1. Read coding Standards Document (provided as part of this lab)
2. Follow the Coding Conventions for your lab tasks.
   - Meaningful variable, function, and class names
   - Well indented
   - Clear modular structure
   - Provide clear and meaningful error messages
   - Push interfaces and declarations up and implementation down
   - Proper Documentation
   - Use appropriate white-space in your programs, and do so in a consistent fashion to make them easy to read.
   - Good coding convention makes program readable and easy to debug
   - Adhere to the same coding style

Task 02: [20 Marks]

You have studied complex numbers in your elementary classes. Recall that complex numbers are of the form $x+yi$ where x and y are real numbers and i is the imaginary unit equal to $\sqrt{-1}$ and $i^2 = -1$.
You are required to create an ADT, Complex Numbers such that the class should contain two data members, real and imaginary, both of type double. It should also have the following member functions:

1. **Complex():** The default constructor that sets both real and imaginary to zero.
2. **Complex(double r):** An overloaded constructor that sets real to *r* and imaginary to zero.
3. **Complex(double r, double i):** Another overloaded constructor that sets real to *r* and imaginary to *i*.
4. **Complex(Complex c):** The copy constructor
5. **Complex add(Complex c):** This function adds the complex numbers and returns a new complex number that represents their sum.
6. **Complex subtract(Complex c):** This function subtracts the two complex numbers.
7. **Complex multiply(Complex c):** This function multiplies the two complex numbers.
8. **Complex divide(Complex c):** This function divides the complex number c by this.
9. **Complex conjugate(Complex c):** This function returns the complex conjugate of c. The complex conjugate of a complex number $z = x + iy$ is defined to be $z = x - iy$.
10. **void print():** The function that prints the complex number object. For eg., if, for the given object, the value of real is 2.4 and that of imaginary is 3.7, this function should print 2.4 + 3.7i on the console.
11. Apart from these functions, you should also define the getter and setter functions: **getReal**, **getImag**, **setReal** and **setImag**.

Test the program by writing the main() function. You should always implement destructors in case of dynamic memory. After coding the program, test your program for various inputs to make sure your program is correct

Task 03: [10 Marks]

Wouldn't it be nice, if instead of adding two Complex Numbers x and y to get z with an add() function like this: z = add(x, y), we could add them in the natural way like this: z = x + y. The great thing about C++ is that you can do this using operator overloading. So now overload the (+) and (-) operator as *friend* functions of the Complex ADT you created in Task – 1.

Task 04: [10 Marks]

Create an ADT, Set. Use the standard mathematics definition of set and include the following operations, you must give description for these operations according to your own understanding in ADT and show to your TAs.

- Create
- Insert
- Remove
- IsMember
- Union
- Intersection
- IsSubset

Task 05: (Home Work) [10 Marks]

Modify the Task 2 by using dynamic memory allocation to a set. Since Set can represent group of any type of data e.g. set of integers, set of employees etc. Use templates to provide user facility of making set of any type of objects.