

Course Name: Data Structures & Algorithm	Course Code: CS-261
Assignment Type: Lab	Dated: 5/9/22
Semester: 3rd	Session: 2022
Lab/Project/Assignment #: 1	CLOs to be covered: 1
Lab Title: Refresher C++	Teacher Name: Waqas Ali

Lab Evaluation:

CLO1	CLO Description/Text					
Levels (Marks)	Level1	Level2	Level3	Level4	Level5	Level6
Psychomotor (5)						
Report Marks (2)						
Total						/50

Rubrics for Current Lab (Optional):

Scale	Marks	Level	Rubric
Excellent	5	L1	Implemented all Tasks by creating Relevant class
Very Good	4	L2	Implemented all tasks with necessary functions
Good	3	L3	Implemented two of the required functions with full functionality.
Basic	2	L4	Implemented two of the required functions with basic functionality.
Barely Acceptable	1	L5	Implemented only one functionality.
Not Acceptable	0	L6	Did not attempt

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.
3. **Anyone caught in act of plagiarism would be awarded an “F” grade in this Lab.**

Task 01:

[20 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 1: Support Code

```
#include<iostream>
#include<cmath>
using namespace std ;

class ComplexNumbers
{
    double realNumber ;
    double imaginaryNumber ;

public:
    ComplexNumbers()
    {
        realNumber = imaginaryNumber = 0 ;
    }

    ComplexNumbers(double realNumber)
    {
        this->realNumber = realNumber ;
        imaginaryNumber = 0 ;
    }

    ComplexNumbers(double realNumber , double imaginaryNumber)
    {
        this->realNumber = realNumber ;
        this->imaginaryNumber = imaginaryNumber ;
    }

    ComplexNumbers(const ComplexNumbers & obj)
    {
        realNumber = obj.realNumber ;
        imaginaryNumber = obj.imaginaryNumber ;
    }
}
```

Your Code is Here.....

Task 02: Your Good Friends → Pointers

1. Write a program to print the address of a variable whose value is input from use.

Your code is here.....

2. Write a program to print the value of the address of the pointer to a variable whose value is input from user.

Your code is here.....

3. Write a function which will take pointer and display the number on screen. Take number from user and print it on screen using that function.

Your code is here...

4. Write a program to find out the greatest and the smallest among three numbers using pointers.

Task 03(HomeWork)

[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 02: Support Code

Task 03 (Home Work)

[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: $z = \text{add}(x, y)$, 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.