UEE 1303(1069): Object-Oriented Programming

Lab #2: C/C++ Overview

In this laboratory session you will:

- Learn how to compile and link multiple files
- Learn how to use the namespace
- Review the basic concept of C/C++ programming

Lab 2-1: Compile and Link multiple Files

✓ Please execute the program `lab2-1`

```
// lab2-1.h
// function prototype, declaration (in header files)
typedef struct {
    double real;
    double image;
} Cplex;
const double pi 3.14159
void showComplex(const Cplex &m);
```

```
// lab2-1.cpp
// function definition (in source files)
#include <iostream>
#include "lab2-1.h"
using namespace std;

void showComplex(const Cplex &m)
{
    cout << m.real;
    if (m.image < 0)
        cout << m.image << "i" << endl;
    else
        cout << "+" << m.image << "i" << endl;
}
```

```
// lab2-1-main.cpp
```

```
// main function, client program
#include <iostream>
#include "lab2-1.h"

int main()
{
    Cplex n;
    n.real = 1 * pi;
    n.image = -0.5;
    showComplex(n);

    return 0;
}
```

- How to Compile?

```
> ls
lab2-1.cpp lab2-1.h lab2-1-main.cpp
> g++ -c lab2-1.cpp
> g++ -c lab2-1-main.cpp
> g++ -o lab2-1  lab2-1.o lab2-1-main.o
> ls
lab2-1 lab2-1.cpp lab2-1.h lab2-1.o lab2-1-main.cpp
lab2-1-main.o
> ./lab2-1
3.14159-0.5i
```

Lab 2-2: Namespaces

✓ Please execute the program `lab2-2` and identify the scope of variable defined in namespace Complex

```
// lab2-2.h
// function prototype, declaration (in header files)
namespace Complex {
    typedef struct {
        double real;
        double image;
    } Cplex;
    const double pi 3.14159
    void showComplex(const Cplex &m);
```

```
}
```

```
// lab2-2.cpp
// function definition (in source files)
#include <iostream>
#include "lab2-2.h"
using namespace std;

namespace Complex {
    void showComplex(const Cplex &m)
    {
        cout << m.real;
        if (m.image < 0)
            cout << m.image << "i" << endl;
        else
            cout << "+" << m.image << "i" << endl;
    }
}
```

```
// lab2-2-main.cpp
// main function, client program
#include <iostream>
#include "lab2-2.h"

int main()
{
    Complex::Cplex n;
    n.real = 1 * Complex::pi;
    n.real = 1 * pi;
    n.image = -0.5;
    Complex::showComplex(n);

    return 0;
}
```

- Please modify the compiling error.

✓ Please modify `lab2-2-main.cpp` as follows and execute the program again.

```
// lab2-2-main.cpp
// main function, client program
#include <iostream>
#include "lab2-2.h"
using namespace Complex;

int main()
{
    Cplex n;
    n.real = 1 * pi;
    n.real = 1 * pi;
    n.image = -0.5;
    showComplex(n);

    return 0;
}
```

- Can you point out the differences between the above programs?

Exercise 2-1

✓ Please write a C++ program to perform the arithmetic operations for complex numbers. You have to read two complex numbers and output the arithmetic results to the console.

✓ Type the following command to execute the program:

```
> ./ex1-1
First complex number: 1.5+6i
Second complex number: -2-10i
==============================
The output results:
A + B = -0.500-4.000i
A - B = 3.500+16.000i
A * B = 57.000-27.000i
A / B = -0.606+0.029i
```

The representation of complex number is `a+bi`, where a means the real part and `b` means the imaginary part. If `a` is equal to zero, it can be written as `0+bi` instead of `bi`. For the same reason, it should be `a+0i` if the complex number has no imaginary part.

✓ Requirement

You have to complete the exercise using the data structure `Cplex` defined in `lab2-1.h` and write three functions: `readComplex()`, `complexOperation()` and `printComplex()`.

The data structure `Cplex` should be defined in `ex2-1.h` and the function should be written in `ex2-1.cpp`.

The `ex2-1-main.cpp` has the content as: (some headers should be added)

```
// ex2-1-main.cpp
int main(int argc, char *argv[])
{
    Cplex a, b;
    readComplex(a, b); // process text file
    // store the results of diff. operation
    Cplex results[4];
    results[0] = complexOperation(a, b, '+');
    results[1] = complexOperation(a, b, '-');
    results[2] = complexOperation(a, b, '*');
    results[3] = complexOperation(a, b, '/');
    printComplex(results);

    return 0;
}
```