UEE1303 S18: Object-Oriented Programming

Function Template & C/C++ Overview/ Pointer

What you will learn from Lab 2

In this laboratory session you will:

1. Learn how to compile and link multiple files
2. Learn how to use the namespace
3. Learn the concept of pointer
4. Review the basic concept of C/C++ programming

LAB 2-1: FUNCTION TEMPLATE

✓ Execute the following program lab2-1, you should edit two files: lab2-1.cpp and maximum.h

```
// lab2-1.cpp
#include <iostream>
#include "maximum.h"
using namespace std;

int main()
{
        int int1, int2, int3;
        cout << "Input three integers: ";
        cin >> int1 >> int2 >> int3;
        cout << "Maximum is " << maximum(int1, int2, int3) << endl;

        double double1, double2, double3 ;
        cout << "Input three double variables: ";
        cin >> double1 >> double2 >> double3;
        cout << "Maximum is " << maximum(double1, double2, double3) << endl;

        char char1, char2, char3 ;
        cout << "Input three characters: ";
        cin >> char1 >> char2 >> char3;
        cout << "Maximum is " << maximum(char1, char2, char3) << endl;

        return 0;
}
```

```
// maximum.h
template <class T>
T maximum(T value1, T value2, T value3)
{
        T max = value1;

        if (value2 > max)
            max = value2;
        if (value3 > max)
            max = value3;

        return max;
}
```

Lab 2-2: Compile and Link multiple Files

✓ Please execute the program lab2-2

```
// lab2-2.cpp
// function definition (in source files)
#include <iostream>
#include "lab2-2.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.h
// function prototype, declaration (in header files)
typedef struct {
    double real;
    double image;
} Cplex;
const double pi = 3.14159;
void showComplex(const Cplex &m);
```

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

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

    return 0;
}
```

✓ How to compile?

```
> ls
lab2-2.cpp lab2-2.h lab2-2-main.cpp
> g++ -c lab2-2.cpp
> g++ -c lab2-2-main.cpp
> g++ -o lab2-2    lab2-2.o lab2-2-main.o
> ls
lab2-2 lab2-2.cpp lab2-2.h lab2-2.o lab2-2-main.cpp lab2-2-main.o
> ./lab2-2
3.14159-0.5i
```

LAB 2-3: NAMESPACES

✓ Please execute the program lab2-3 and identify the scope of variable defined in namespace Complex

```
// lab2-3.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);
}
```

```
// lab2-3.cpp
// function definition (in source files)
#include <iostream>
#include "lab2-3.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;
    }
}
```

```
// lab2-3-main.cpp
// main function, client program
#include <iostream>
#include "lab2-3.h"

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

        return 0;
}
```

Lab 2-4: Pointer

✓ Program `lab2-4` below shows some examples of using for pointer manipulation including pointer declarations and assignments.

```
// lab2-4-1.cpp
#include <iostream>
using namespace std;
int main()
{
        double a = 1.34;
        double *pa = &a;
        cout << "a = " << a << endl;
        cout << "&a = " << &a << endl;
        cout << "*a = " << *a << endl;
        cout << "pa = " << pa << endl;
        cout << "&pa = " << &pa << endl;
        cout << "*pa = " << *pa << endl;
        *pa = 6.5;
        cout << "a = " << a << endl;
        cout << "*pa = " << *pa << endl;
        return 0;
}
```

✓ Please try to explain the execution results by yourself. Notice that there is a compiler error in this example.

✓ The following is an example to use pointer arithmetic to dereference the array elements.

```
// lab2-4-2.cpp
#include <iostream>
#include <cstdlib>

using std::cout;
using std::endl;

int main()
```

```
{
    int a[10];
    srand(time(NULL));

    for (int i = 0;i < 10; i++)
        a[i] = rand()%20 + 10;

    int *pa = a;
    for (int i = 0; i < 10; i++)
        cout << *(pa++) << " ";
    cout << endl;

    return 0;
}
```

- ✓ The program demonstrates that a pointer is used to point the structure object.

```
// lab2-4-3.cpp
#include <iostream>
typedef struct
{
    int x;
    int y;
    double value;
}Point2D;

void assignPoint2D(Point2D *obj, int x, int y, double value)
{
    obj->x = x;
    obj->y = y;
    obj->value = value;
}

void displayPoint2D(Point2D *obj)
{
    std::cout << "(" << obj->x << "," << obj->y << ") = "
<< obj->value << std::endl;
}
int main()
{
    Point2D ptArray[10];
    for (int i = 0; i < 10; i++)
    {
        assignPoint2D(ptArray[i], i, i+2, i*10);
        displayPoint2D(ptArray[i]);
    }

    return 0;
}
```

- ✓ Please fix the compiler error.

EXERCISE 2-1: THE AREA OF THE CIRCLE

✓ Write a function template selectionSort. Write a driver program that inputs, sorts and outputs an int array and a float array.
The int array should be
int a[SIZE] = { 2, 9, 10, 1, 7, 3, 4, 5, 8, 6 };
The float array should be
double b[SIZE] = { 2.2, 9.9, 10.1, 1.1, 7.7, 3.3, 4.4, 5.5, 8.8, 6.6};

```
>./ex2-1
int data items in original order
 2      9     10      1      7      3      4      5      8      6
int data items in ascending order
1      2      3      4      5      6      7      8      9     10
float data items in original order
2.2    9.9   10.1    1.1    7.7    3.3    4.4    5.5    8.8    6.6
float point data items in ascending order
1.1    2.2    3.3    4.4    5.5    6.6    7.7    8.8    9.9   10.1
>
```

EXERCISE 2-2: COMPLEX ARITHMETIC

✓ 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:

```
> ./ex1-2
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 `lab2-2.h` and write two functions: `complexOperation()` and `printComplex()`.
The data structure `Cplex` should be defined in `ex2-2h` and the function should be written in `ex2-2.cpp.`
The `ex2-2-main.cpp` has the content as: (some headers should be added)

```
// ex2-2-main.cpp
int main(int argc, char *argv[])
{
        Cplex a, b;
        // promotes the user to input data
        ………
        // 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;
}
```