Object Oriented Programming (IGS2130)

Lab 8

Instructor:
Choonwoo Ryu, Ph.D.

INHA UNIVERSITY

Exercise #1

- Write a class named Calculate so that the main function below can output as like the execution example.

```
int main(void) {
    Calculate cc;
    int i;
    char op;
    double value;
    for (i = 0; i < 10; ++i) {
        cout << "Select math operator(+,-,*,/): ";
        cin >> op;
        cout << "Enter a real number for the math: ";
        cin >> value;

        switch (op) {
        case '+':
            cout << cc.getValue() << " + " << value;
            cout << " = " << cc.add(value) << endl;
            break;
        case '-':
            cout << cc.getValue() << " - " << value;
            cout << " = " << cc.substract(value) << endl;
            break;
        case '*':
            cout << cc.getValue() << " * " << value;
            cout << " = " << cc.multiply(value) << endl;
            break;
        case '/':
            cout << cc.getValue() << " / " << value;
            cout << " = " << cc.divide(value) << endl;
            break;
        }
    }
    return 0;
}
```

```
Select math operator(+,-,*,/): +
Enter a real number for the math: 10.5
0 + 10.5 = 10.5
Select math operator(+,-,*,/): -
Enter a real number for the math: 7.5
10.5 - 7.5 = 3
Select math operator(+,-,*,/): *
Enter a real number for the math: 3.5
3 * 3.5 = 10.5
Select math operator(+,-,*,/): /
Enter a real number for the math: 0.5
10.5 / 0.5 = 21
Select math operator(+,-,*,/): -
Enter a real number for the math: 10
21 - 10 = 11
Select math operator(+,-,*,/): *
Enter a real number for the math: 5.5
11 * 5.5 = 60.5
Select math operator(+,-,*,/): -
Enter a real number for the math: 0.5
60.5 - 0.5 = 60
Select math operator(+,-,*,/): /
Enter a real number for the math: 6
60 / 6 = 10
Select math operator(+,-,*,/): +
Enter a real number for the math: 8
10 + 8 = 18
Select math operator(+,-,*,/): /
Enter a real number for the math: 2
18 / 2 = 9
```

Exercise #2

- Improve the Calculate class in Ex#01 so that it can undo up to the last five math operations. The main function below can output as like the execution example.

```
int main(void) {
    Calculate cc;

    ……

    cout << endl << endl;
    for (i = 0; i < 10; ++i) {
        cout << "Stored the last math operation: ";
        flag = cc.lastOperation(op, value);
        if (!flag)
            cout << "None" << endl;
        else
            cout << op << ", " << value << endl;

        if (flag) {
            cc.undo();
            cout << "Undo the last math operation..." << endl;
            cout << "Value inside the class object: ";
            cout << cc.getValue() << endl;
        }
    }
    cout << "Value inside the class object: ";
    cout << cc.getValue() << endl;

    return 0;
}
```

```
Select math operator(+,-,*,/): +
Enter a real number for the math: 10.5
0 + 10.5 = 10.5
Select math operator(+,-,*,/): -
Enter a real number for the math: 7.5
10.5 - 7.5 = 3
Select math operator(+,-,*,/): *
Enter a real number for the math: 3.5
3 * 3.5 = 10.5
Select math operator(+,-,*,/): /
Enter a real number for the math: 0.5
10.5 / 0.5 = 21
Select math operator(+,-,*,/): -
Enter a real number for the math: 10
21 - 10 = 11
Select math operator(+,-,*,/): *
Enter a real number for the math: 5.5
11 * 5.5 = 60.5
Select math operator(+,-,*,/): -
Enter a real number for the math: 0.5
60.5 - 0.5 = 60
Select math operator(+,-,*,/): /
Enter a real number for the math: 6
60 / 6 = 10
Select math operator(+,-,*,/): +
Enter a real number for the math: 8
10 + 8 = 18
Select math operator(+,-,*,/): /
Enter a real number for the math: 2
18 / 2 = 9
```

```
Stored the last math operation: /, 2
Undo the last math operation...
Value inside the class object: 18
Stored the last math operation: +, 8
Undo the last math operation...
Value inside the class object: 10
Stored the last math operation: /, 6
Undo the last math operation...
Value inside the class object: 60
Stored the last math operation: -, 0.5
Undo the last math operation...
Value inside the class object: 60.5
Stored the last math operation: *, 5.5
Undo the last math operation...
Value inside the class object: 11
Stored the last math operation: None
Stored the last math operation: None
Stored the last math operation: None
Stored the last math operation: None
Stored the last math operation: None
Value inside the class object: 11
```

Exercise #3

- Implement a copy constructor of the Calculate class so that the final calculation result is copied, but the past operation history is not copied. The main function below can output as like the execution example.

```
int main(void) {
    Calculate cc;

    ……

    Calculate dd{ cc };

    cout << endl << endl;
    for (i = 0; i < 10; ++i) {
        cout << "Stored the last math operation: ";
        flag = dd.lastOperation(op, value);
        if (!flag)
            cout << "None" << endl;
        else
            cout << op << ", " << value << endl;

        if (flag) {
            dd.undo();
            cout << "Undo the last math operation..." << endl;
            cout << "Value inside the class object: ";
            cout << dd.getValue() << endl;
        }
    }
    cout << "Value inside the class object: ";
    cout << dd.getValue() << endl;

    return 0;
}
```

```
Select math operator(+,-,*,/): +
Enter a real number for the math: 10.5
0 + 10.5 = 10.5
Select math operator(+,-,*,/): -
Enter a real number for the math: 7.5
10.5 - 7.5 = 3
Select math operator(+,-,*,/): *
Enter a real number for the math: 3.5
3 * 3.5 = 10.5
Select math operator(+,-,*,/): /
Enter a real number for the math: 0.5
10.5 / 0.5 = 21
Select math operator(+,-,*,/): -
Enter a real number for the math: 10
21 - 10 = 11
Select math operator(+,-,*,/): *
Enter a real number for the math: 5.5
11 * 5.5 = 60.5
Select math operator(+,-,*,/): -
Enter a real number for the math: 0.5
60.5 - 0.5 = 60
Select math operator(+,-,*,/): /
Enter a real number for the math: 6
60 / 6 = 10
Select math operator(+,-,*,/): +
Enter a real number for the math: 8
10 + 8 = 18
Select math operator(+,-,*,/): /
Enter a real number for the math: 2
18 / 2 = 9
```

```
Stored the last math operation: None
Stored the last math operation: None
Stored the last math operation: None
Stored the last math operation: None
Stored the last math operation: None
Stored the last math operation: None
Stored the last math operation: None
Stored the last math operation: None
Stored the last math operation: None
Stored the last math operation: None
Value inside the class object: 9
```

Exercise #4

- Create a class named Time so the following main() function outputs as like the execution result below. Note that the printTime() member function prints the time of object in 24-hour clock format if the object is a non-const object otherwise prints it in 12-hour clock format.

```
int main() {
    const Time t0{ 7, 3, 5 };
    Time t1{ 13, 45, 9 };
    const Time t2{ t1 };

    cout << "t0: ";
    t0.printTime();
    cout << "t1: ";
    t1.printTime();
    cout << "t2: ";
    t2.printTime();

    return 0;
}
```

```
t0: 07:03:05 AM
t1: 13:45:09
t2: 01:45:09 PM
```

Hint.

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

int main() {
    cout << "##";
    cout << setfill('0') << setw(5) << 123;
    cout << "##";

    return 0;
}
```

```
##00123##
```

Exercise #5

- Write a setArray() function to set the values of elements of the IntArray class's array data and a displayArray() function to output the array values of the class. The functions must be friend functions of the IntArray class and have direct access to private members.

```
#include<iostream>
#include<iomanip>
#include<ctime>
#include<cstdlib>
using namespace std;
class IntArray
{
private:
    int m_len{ 0 };
    int* m_data{ nullptr };

public:
    IntArray(int len)
        : m_len{ len }
    {
        m_data = new int[m_len];
    }
    ~IntArray() {
        if (m_data) delete[] m_data;
    }
};
```

```
const int arSize = 20;

int main() {
    int i;
    int data[arSize];
    IntArray ar{ arSize };

    srand((unsigned int)time(NULL));
    for (i = 0; i < arSize; ++i) data[i] = rand() % 100;

    setArray(ar, data, arSize);
    displayArray(ar);

    return 0;
}
```

```
[ 0] 73
[ 1] 16
[ 2] 4
[ 3] 37
[ 4] 36
[ 5] 58
[ 6] 89
[ 7] 46
[ 8] 15
[ 9] 29
[10] 85
[11] 32
[12] 93
[13] 34
[14] 53
[15] 59
[16] 76
[17] 59
[18] 9
[19] 26
```