Object Oriented Programming (IGS2130)

Lab 7

Instructor:
Choonwoo Ryu, Ph.D.

INHA UNIVERSITY

Exercise #1

- Create and use the Rectangle class so that the main() function below can produce the same result as the given output.
  - setWidth(): set the width of the rectangle
  - setHeight(): set the height of the rectangle
  - Display(): Print the width and height of the rectangle
  - Use this pointer

```
int main() {
    Rectangle rect;
    rect.Display();
    rect.setWidth(10).setHeight(20).Display();

    return 0;
}
```

```
Width = 0 Height = 0
Width = 10 Height = 20
```

Exercise #2

- Executing the given program below generates a runtime error. Implement the copy constructor in the IntArray class so that it can operate like the given output without runtime error.

```
#include<iostream>
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;
    }
    void set(int index, int value) {
        if (index >= 0 && index < m_len)
            m_data[index] = value;
    }
    int get(int index, int err) const {
        if (index >= 0 && index < m_len)
            return m_data[index];
        else
            return err;
    }
};
```

```
int main() {
    int i;
    cout << "=== IntArray a{ 10 } ===" << endl;
    IntArray a{ 10 };
    for (i = 0; i < 10; ++i)
        a.set(i, i * 10 + 5);
    cout << "a: ";
    for (i = 0; i < 10; ++i)
        cout << a.get(i, -1) << ' ';
    cout << endl;

    cout << "=== IntArray b{ a } ===" << endl;
    IntArray b{ a };
    cout << "b: ";
    for (i = 0; i < 10; ++i)
        cout << a.get(i, -1) << ' ';

    return 0;
}
```

```
=== IntArray a{ 10 } ===
a: 5 15 25 35 45 55 65 75 85 95
=== IntArray b{ a } ===
b: 5 15 25 35 45 55 65 75 85 95
```

Exercise #3: OOP Project: Step 02

- Upgrade our non-OOP based bank application (ver 0.1) to version 0.2
  - Create Account class instead of using Account structure
  - Apply information hiding
  - Use constructor and destructor

- Additional changes
  - Use char * and dynamic memory allocation for the member variable of the customer's name in Account class
  - Use pointer array of account class to store multiple accounts in the program

To start easier, use a single file in this project.

Exercise #3: OOP Project: Step 02

- Account class definition

```
class Account {
private:
    int m_accID;
    int m_balance;
    char * m_cusName;

public:
    Account(int ID, int balance, char *cname) {
         // implementation required ....
    }
    ~Account() {
        // implementation required ....
    }
    int GetAccID(void) {
        // implementation required ....
    }
    void Deposit(int money) {
        // implementation required ....
    }
    int Withdraw(int money) {
        // implementation required ....
    }
    void ShowAccInfo(void) {
        // implementation required ....
    }
};
```

Exercise #3: OOP Project: Step 02

- Main function and global variables

```
Account *accArr[MAX_ACC_NUM]; // Account array
int accNum = 0;       // # of accounts

int main(void) {
    int choice, i;

    while (1) {
        ShowMenu();
        cout << "Select menu: ";
        cin >> choice;
        cout << endl;

        switch (bank(choice)) {
        case bank::MAKE:
            MakeAccount();
            break;
        case bank::DEPOSIT:
            DepositMoney();
            break;
        case bank::WITHDRAW:
            WithdrawMoney();
            break;
        case bank::INQUIRE:
            ShowAllAccInfo();
            break;
        case bank::EXIT:
            for (i = 0; i < accNum; i++)
                delete accArr[i];
            return 0;
        default:
            cout << "Illegal selection.." << endl;
        }
    }
    return 0;
}
```

Exercise #3: OOP Project: Step 02

- Account class handling functions
  - Same function prototype but…
  - Must be modified according to the changes in the Account class definition.

```
void ShowMenu(void);
void MakeAccount(void);
void DepositMoney(void);
void WithdrawMoney(void);
void ShowAllAccInfo(void);
int GetAccIdx(int);
```

Hint.

```
void MakeAccount(void) {
   .....
   .....
   accArr[accNum] = new Account{id, balance, name};
   accNum++;
}
```