Object Oriented Programming (IGS2130)

Lab 6

Instructor:
Choonwoo Ryu, Ph.D.

INHA UNIVERSITY

Exercise #1

- Point3d class
  - Write a class named Point3d. Point3d should contain three member variables of type double: m_x, m_y, and m_z defaulted to 0.0s. Provide a constructor and a print member function.
  - The following program should run:

```
int main() {
  Point3d p1{};
  Point3d p2{ 3.0, 4.0, 5.0 };
  p1.print();
  p2.print();

  return 0;
}
```

  - This should print:

```
Point3d(0, 0, 0)
Point3d(3, 4, 5)
```

Exercise #2

- Upgrade Exercise #1
  - Add a member function named distanceTo that takes another Point3d as a parameter, and calculates the distance between them.

$$d = ((x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2)^{1/2}$$

  - The following program should run:

```
int main() {
  Point3d p1{};
  Point3d p2{ 3.0, 4.0, 5.0 };
  p1.print();
  p2.print();
  cout << "Distance: " << p1.distanceTo(p2) << "\n";
  return 0;
}
```

  - This should print:

```
Point3d(0, 0, 0)
Point3d(3, 4, 5)
Distance: 7.07107
```

Exercise #3: OOP Project: Step 03

- Upgrade our bank application version 0.2 to 0.3
  - Improve Account class
    - Create a copy constructor to support deep copy
    - Convert all possible member functions into const member functions
  - Use the strcpy_s() function instead of strcpy(), to get rid of the error message in Visual Studio

Account class

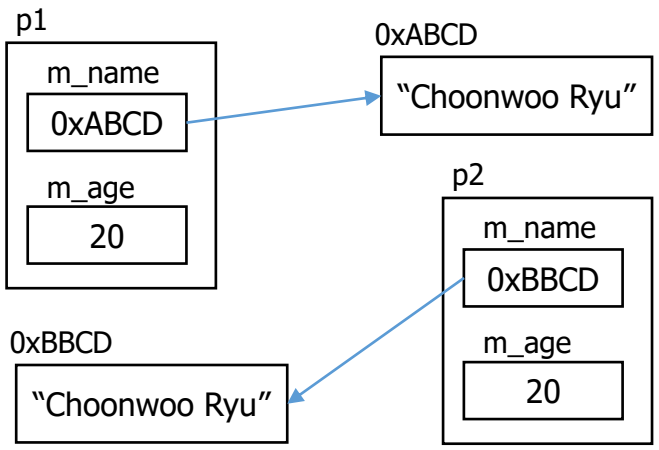
```
class Account {
private:
    int m_accID;
    int m_balance;
    char* m_cusName;

public:
    Account(int ID, int balance, char* cname)
        : m_accID{ ID }, m_balance{ balance }
    {
        int len = strlen(cname) + 1;
        m_cusName = new char[len];
        strcpy_s(m_cusName, len, cname);
    }
    ~Account() {
        delete[]m_cusName;
    }
    int GetAccID(void) {
        return m_accID;
    }
    void Deposit(int money) {
        if (money > 0)
            m_balance += money;
    }
    int Withdraw(int money) {
        if ((money < 0) || (money > m_balance))
            return -1;
        m_balance -= money;
        return money;
    }
    void ShowAccInfo(void) {
        cout << "Account ID: " << m_accID
<< endl;
        cout << "Name: " << m_cusName << endl;
        cout << "Balance: " << m_balance
<< endl << endl;
    }
};
```

Example of copy constructor

Deep copy

```
int main(){
    Person p1("Choonwoo Ryu", 20);
    Person p2{ p1 };
    cout << "p1: "; p1.showInfo();
    cout << "p2: "; p2.showInfo();
    return 0;
}
```

```
p1: Name: Choonwoo Ryu, Age: 20
p2: Name: Choonwoo Ryu, Age: 20
before delete[]m_name;
after delete[]m_name;
before delete[]m_name;
after delete[]m_name;
```




```
#include <iostream>
using namespace std;
class Person {
private:
    char* m_name;
    int m_age;
public:
    Person(const char *name, int age):
        m_age{age}
    {
        m_name = new char[strlen(name) + 1];
        strcpy(m_name, name);
    }
    Person(const Person& cp) :
        m_age{ cp.m_age }
    {
        m_name = new char[strlen(cp.m_name) + 1];
        strcpy(m_name, cp.m_name);
    }
    ~Person() {
        cout << "before delete[]m_name;\n";
        delete[]m_name;
        cout << "after delete[]m_name;\n";
    }
    void showInfo(void) {
        cout << "Name: " << m_name;
        cout << ", Age: " << m_age << endl;
    }
};
```

Example of const member function

- Const member functions
  - Will not modify the object or call any non-const member functions
  - const class objects can only explicitly call const member functions

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

class Something {
public:
    int m_value;
    Something() : m_value{ 0 } { }
    void setValue(int value) { m_value = value; }
    // const member function
    int getValue() const { return m_value; }
};

int main() {
    // calls default constructor
    const Something something{};
    cout << something.getValue() << endl;

    return 0;
}
```

add const keyword after parameter list, but before function body

```
0
```

strcpy_s()

strcpy_s(char *dest, rsize_t dest_size, const char *src);

dest: the destination string buffer
dest_size: Size of the destination string buffer in char units
src: Null-terminated source string buffer