

C++ 프로그래밍

실습 10주차

* 프로젝트 작성시 주의사항

1. 프로젝트 이름과 소스 파일 이름은 동일하게 작성한다.
이름은 학번+P실습번호로 한다.
예) 20089999P10_1, 20089999P10_2
2. 제출 요령
 - 각 프로젝트의 Debug폴더와 Database 파일을 삭제한다.
 - 실습 전체 프로젝트를 압축하여 제출한다.

10-1. 다음 요구사항에 맞는 프로그램을 작성하고 실행해보시오. 다음 프로그램의 실행결과를 참조해서 _____에 알맞은 코드를 작성하여 보시오.

* 프로그램 요구사항

- 프렌드 함수는 클래스의 public 정의부에 함수의 원형만 정의하고 함수의 구현부는 클래스의 외부에서 정의하거나 또는 함수전체를 클래스의 내부에서 정의한다.

* 실행 결과

객체 멤버함수로 (4,5) Length = 6.40312
프렌드 함수로 (4,5) Length = 6.40312

```
#include <iostream>
#include <math.h>
using namespace std;

class Point
{
    int m_x, m_y;
    double Length( )
    {
        return ( sqrt(double(m_x * m_x + m_y * m_y)) );
    }
public:
    _____ // 매개변수가 두 개인 생성자 함수
    {
        m_x = x;
        m_y = y;
    }
}
```

```

void ShowPoint();
_____ void ShowMe(Point &p);    // 프렌드 함수로 정의
};

void Point::ShowPoint()
{
    cout << "객체 멤버함수로 (" << m_x << ',' << m_y << ')' <<
        "    Length = " << _____ << endl;    // length 함수 호출
}

void ShowMe(Point &p)    // 프렌드 함수 구현(객체의 내용을 출력)
{
    cout << "프렌드 함수로 (" << p.m_x << ',' << p.m_y << ')' <<
        "    Length  = " << _____ << endl;    // length 함수 호출
}

int main()
{
    _____;    // Point 클래스로 obj 라는 객체 선언(초기 값 4, 5)
    _____;    // 객체 멤버 함수를 사용하여 객체의 내용을 출력
    _____;    // 프렌드 함수를 사용하여 객체의 내용을 출력

    system("PAUSE");
    return 0;
}

```

10-2. 다음 요구사항에 맞는 프로그램을 작성하고 실행해보시오. _____부분에 알맞은 코드를 채워 넣고 실행하여 프로그램이 잘 작성되었는지 실행결과와 비교해 보시오.

*** 프로그램 요구사항**

- Point 클래스를 정의하고 비공개 변수들을 더하고 치환하는 연산자 함수를 중복한다. 연산자 함수는 멤버 함수로 정의한다.

*** 실행 결과**

ob3의 x 는 9 y 는 11
ob4의 x 는 5 y 는 4
계속하려면 아무 키나 누르십시오 . . .

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

class Point {
    int m_x, m_y;
public :
    // 매개변수가 있는 생성자 함수 (두 매개변수의 디폴트 인수로 0을 사용)
    _____;
    void Get_XY(int &a, int &b);
    Point operator+(Point ob2);
    Point operator=(Point ob2);
};

Point::Point(int x, int y)
{
    m_x = x;
    m_y = y;
}

void Point::Get_XY(int &a, int &b)
{
    a = m_x;
    b = m_y;
}
```

```

// Point 클래스의 비공개 변수들을 더하는 기능을 가지도록 + 연산자를 중복한다.
_____

{
    Point temp;
    _____;
    _____;
    _____;
}

// Point 클래스의 비공개 변수들을 치환하는 기능을 가지도록 = 연산자를 중복한다.
_____

{
    _____;
    _____;
    _____;           // 치환된 객체를 반환한다.
}

int main()
{
    Point ob1(5, 4), ob2(4, 7), ob3, ob4;
    int x, y;

    _____;           // ob1과 ob2를 더해서 ob3에 치환한다.
    _____;           // Get_XY() 함수를 이용하여 ob3의 x, y 값을 얻어온다
    cout << "ob3 의 x 는 " << x << " y 는 " << y << "\n";

    _____;           // ob1을 ob2에 ob2를 ob3에 ob3를 ob4에
                           // 연속적으로 치환한다.(한 줄로 구현)
    _____;           // Get_XY() 함수를 이용하여 ob4의 x, y 값을 얻어온다
    cout << "ob4 의 x 는 " << x << " y 는 " << y << "\n";

    system("PAUSE");
    return 0;
}

```

10-3. 다음 요구사항에 맞는 프로그램을 작성하고 실행해보시오. _____부분에 알맞은 코드를 채워 넣고 실행하여 프로그램이 잘 작성되었는지 실행결과와 비교해 보시오.

*** 프로그램 요구사항**

- Point 클래스를 정의하고 비공개 변수에 더하는 연산자 함수를 중복한다. 연산자 함수는 프렌드 함수로 정의하고 연산자 함수는 (객체+ 객체), (객체 + int), (int + 객체)의 연산을 수행하도록 + 연산자를 세 번 중복한다.

*** 실행 결과**

ob3의 x 는 9 y 는 11
ob4의 x 는 15 y 는 17
ob5의 x 는 12 y 는 14
계속하려면 아무 키나 누르십시오 . . .

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

class Point {
    int m_x, m_y;
public :
    // 매개변수가 있는 생성자 함수 (두 매개변수의 디폴트 인수로 0을 사용)
    _____;
    void Get_XY(int &a, int &b);
    friend Point operator+(Point ob1, Point ob2);
    friend Point operator+(Point ob1, int i);
    friend Point operator+(int i, Point ob1);
};

Point::Point(int x, int y)
{
    m_x = x;
    m_y = y;
}

void Point::Get_XY(int &a, int &b)
{
    a = m_x;
    b = m_y;
}
```

```

// Point 클래스의 객체들을 더하는 기능을 가지도록 +를 중복한다. (객체 + 객체)
{
    Point temp;
    _____;
    _____;
    _____;
}

// Point 클래스의 객체에 int 변수(값)을 더하는 기능을 가지도록 +를 중복한다. (객체 + int)
{
    Point temp;
    _____;
    _____;
    _____;
}

// Point 클래스의 객체에 int 변수(값)을 더하는 기능을 가지도록 +를 중복한다. (int + 객체)
{
    Point temp;
    _____;
    _____;
    _____;
}

int main()
{
    Point ob1(5, 4), ob2(4, 7), ob3, ob4, ob5;
    int x, y;

    _____; // ob1과 ob2를 더해서 ob3에 치환한다.
    _____; // Get_XY() 함수를 이용하여 ob3의 x, y 값을 얻어온다.
    cout << "ob3 의 x 는 " << x << " y 는 " << y << "\n";

    _____; // ob3에 6을 더해 ob4에 치환한다. (객체 + int)
    _____; // Get_XY() 함수를 이용하여 ob4의 x, y 값을 얻어온다.
    cout << "ob4 의 x 는 " << x << " y 는 " << y << "\n";

    _____; // ob3에 3을 더해 ob5에 치환한다. (int + 객체)
    _____; // Get_XY() 함수를 이용하여 ob5의 x, y 값을 얻어온다.
    cout << "ob5 의 x 는 " << x << " y 는 " << y << "\n";

    system("PAUSE");
    return 0;
}

```

10-4. 다음 요구사항에 맞는 프로그램을 작성하고 실행해보시오. _____ 부분에 알맞은 코드를 채워 넣고 실행하여 프로그램이 잘 작성되었는지 실행결과와 비교해 보시오.

*** 프로그램 요구사항**

- 클래스 Coord에는 좌표 x, y와 원점으로부터 거리를 구하는 멤버함수 Length()가 있다. 이 클래스에 두 점의 거리를 구하고, 비공개 변수에 정수를 곱하고 부호를 바꾸는 연산자 함수를 중복한다. 연산자 함수는 멤버함수로 정의한다.

*** 실행 결과**

대전대 = (3, 4)
시청 = (2, 8)
원점에서 대전대까지 거리 = 5
원점 - 대전대 거리 = 5
대전대 - 시청 거리 = 4.12311
대전대의 3배 좌표 = (9, 12)
대전대 원점 대칭 = (-3, -4)
계속하려면 아무 키나 누르십시오 . . .

```
#include <iostream>
#include <string>
#include <math.h>
using namespace std;

class Coord
{
    int m_x, m_y;                // 두 좌표를 나타내는 비공개 변수
public:
    // 매개변수가 있는 생성자 함수 (두 매개변수의 디폴트 인수로 0을 사용)
    _____
    {
        m_x = x;
        m_y = y;
    }
    int GetX() { return m_x; }
    int GetY() { return m_y; }
    double Length();
    void ShowPoint(string pnt);

    Coord operator-();           // 부호를 바꾸는 연산자 함수
    Coord operator*(int k);      // 정수를 곱하는 연산자 함수
    double operator^(Coord &p); // 두 점의 거리를 구하는 연산자 함수
};
```

```

// 원점으로부터 객체의 거리를 구하는 함수
double Coord::Length()
{
    return sqrt(double(m_x*m_x + m_y*m_y));
}

void Coord::ShowPoint(string pnt)
{
    cout << pnt << "= (" << m_x << ", " << m_y << ')' << endl;
}

// Coord 클래스의 객체의 비공개 변수에 정수를 곱하는 기능을 가지도록 *를 중복한다.
// 정수를 곱한 후에 원래 호출한 객체의 값이 변경되어서는 안 된다.
Coord Coord::operator*(int k)
{
    Coord temp;
    _____;
    _____;
    _____;
}

// Coord 클래스의 객체들의 거리를 구하는 기능을 가지도록 ^를 중복한다.
double Coord::operator^(Coord &p)
{
    double length;
    length = _____;
    return length;
}

// Coord 클래스의 객체의 비공개 변수 부호를 바꾸는 기능을 가지도록 -를 중복한다.
// 부호를 바꾼 후에 원래 호출한 객체의 값이 변경되어서는 안 된다.
Coord Coord::operator-( )
{
    Coord temp;
    _____;
    _____;
    _____;
}

```



```

int main()
{
    // origin 좌표는 원점 (0, 0)이 된다.
    Coord origin, dju(3, 4), cityhall(2, 8);
    dju.ShowPoint("대전대 ");
    cityhall.ShowPoint("시청 ");

    // Length()함수를 이용하여 원점에서 대전대까지의 거리를 구한다.
    cout << "원점에서 대전대까지 거리 = " << _____ << endl;

    // 연산자 함수 ^ 를 이용하여 원점에서 대전대까지의 거리를 구한다.
    cout << "원점 - 대전대 거리 = " << (_____) << endl;

    // 연산자 함수 ^ 를 이용하여 대전대에서 시청까지의 거리를 구한다.
    cout << "대전대 - 시청 거리 = " << (_____) << endl;

    // 연산자 함수 * 를 이용하여 대전대에서 3배되는 좌표를 구해 출력한다.
    (_____.ShowPoint("대전대의 3배 좌표 ");

    // 연산자 함수 - 를 이용하여 대전대에서 원점에 대칭되는 좌표를 구해
    // 출력한다.
    (_____.ShowPoint("대전대 원점 대칭 ");

    system("PAUSE");
    return 0;
}

```