

## 과제 1번

13-1. coord는 다음과 같이 정의된다. 이 클래스와 관련된, \*와 / 연산자를 중복시켜 보고, 이것이 작동하는 예제 프로그램을 작성하여라.

```
class coord {  
private:  
    int x, y; // 좌표 값  
public:  
    coord() { x=0; y=0; }  
    coord(int i, int j) { x=i; y=j; }  
    void get_xy(int &i, int &j) { i=x; j=y; }  
    // 연산자 중복  
};
```

```
1  #include <iostream>  
2  using namespace std;  
3  
4  class coord {  
5  private:  
6      int x, y;  
7  public:  
8      coord() { x = 0; y = 0; }  
9      coord(int i, int j) { x = i; y = j; }  
10     void get_xy(int &i, int &j) { i = x; j = y; }  
11     coord operator*(coord gop);  
12     coord operator/(coord nan);  
13 };  
14  
15 // 연산자 * 중복  
16 coord coord::operator*(coord dif) {  
17     coord temp; // 임시저장을 위한 coord형 변수 선언  
18     temp.x = x * dif.x; // * 연산한 x값을 temp.x에 저장  
19     temp.y = y * dif.y; // * 연산한 y값을 temp.y에 저장  
20     return temp;  
21 }  
22  
23 // 연산자 / 중복  
24 coord coord::operator/(coord nan) {  
25     coord temp; // 임시저장을 위한 coord형 변수 선언  
26     temp.x = x / nan.x; // / 연산한 x값을 temp.x에 저장  
27     temp.y = y / nan.y; // / 연산한 y값을 temp.y에 저장  
28     return temp;  
29 }  
30  
31 int main() {  
32     coord a(8, 4), b(4, 2), result;  
33     int p, q; // get_xy 활용을 위한 변수 선언  
34     // 연산자 * 사용  
35     result = a * b;  
36     result.get_xy(p, q); // result의 x값을 p에 y값을 q에 저장  
37     cout << "result.x : " << p << "    " << "result.y : " << q << endl;  
38  
39     // 연산자 / 사용  
40     result = a / b;  
41     result.get_xy(p, q); // result의 x값을 p에 y값을 q에 저장  
42     cout << "result.x : " << p << "    " << "result.y : " << q << endl;  
43  
44     return 0;  
45 }
```

C:\Users\Owner\Desktop\단국대학교\객체지향프로그래밍\C++SelfStudy\Project1\Debug\Project1.exe

```
result.x : 32    result.y : 8  
result.x : 2     result.y : 2  
계속하려면 아무 키나 누르십시오 . . .
```

## 과제 2번

13-2. coord 클래스와 관련된, ==와 && 연산자를 중복시켜 보고, 이것이 작동하는 예제 프로그램을 작성하여라.

```
C:\Users\Owner\Desktop\한국대학교\객체지향프로그래밍\C++SelfStudy\Project1\Debug#

1  #include <iostream>
2  using namespace std;
3
4  class coord {
5  private:
6      int x, y;
7  public:
8      coord() { x = 0; y = 0; }
9      coord(int i, int j) { x = i; y = j; }
10     void get_xy(int &i, int &j) { i = x; j = y; }
11     bool operator==(coord sam);
12     bool operator&&(coord amd);
13 };
14
15 // 연산자 == 중복
16 bool coord::operator==(coord sam) {
17     if (x == sam.x && y == sam.y) { // 같은지 다른지 검사
18         return true; // 같으면 true
19     }
20     else {
21         return false; // 다르면 false 반환
22     }
23 }
24
25 // 연산자 && 중복
26 bool coord::operator&&(coord amd) {
27     if (x == y && amd.x == amd.y) { // 같은지 다른지 검사
28         return true; // 같으면 true
29     }
30     else {
31         return false; // 다르면 false 반환
32     }
33 }
34
35 int main() {
36     coord a(20, 30), b(20, 30);
37     // ==가 참일때
38     cout << "a == b --> " << (a == b ? "true" : "false") << endl;
39     cout << "a && b --> " << (a && b ? "true" : "false") << '\n' << endl;
40
41     coord c(10, 10), d(20, 20);
42     // &&가 참일때
43     cout << "c == d --> " << (c == d ? "true" : "false") << endl;
44     cout << "c && d --> " << (c && d ? "true" : "false") << endl;
45
46     return 0;
47 }
```

a == b --> true  
a && b --> false  
  
c == d --> false  
c && d --> true  
계속하려면 아무 키나 누르십시오 . . .

### 과제 3번

13-4. coord 클래스에서 --를 중복하여라. 선행(prefix)과 후행(postfix) 표기로 작성해 보아라.

```
C:\Users\Owner\Desktop\단국대학교부객제지향프로그래밍\C++SelfStudy\Project1\Debug\
1  #include <iostream>
2  using namespace std;
3
4  class coord {
5  private:
6      int x, y;
7  public:
8      coord() { x = 0; y = 0; }
9      coord(int i, int j) { x = i; y = j; }
10     void get_xy(int &i, int &j) { i = x; j = y; }
11     coord operator--(); // 선행
12     coord operator--(int notused); // 후행
13 };
14
15 // 연산자 선행 -- 중복
16 coord coord::operator--() {
17     --x; // x의 값 1감소
18     --y; // y의 값 1감소
19     return *this; // 감소된 값을 반환
20 }
21
22 // 연산자 후행 -- 중복
23 coord coord::operator--(int notused) {
24     coord temp = *this; // 원래의 값을 저장
25     x--; // x의 값 1감소
26     y--; // y의 값 1감소
27     return temp; // 원래의 값을 반환
28 }
29
30
31 int main() {
32     coord prefix(100, 200), postfix(100, 200), result;
33     int m, n;
34     // prefix 선행감소
35     result = --prefix;
36     result.get_xy(m, n); //
37     cout << "Before : " << m << ", " << n << endl;
38     prefix.get_xy(m, n);
39     cout << "After : " << m << ", " << n << '\n' << endl;
40
41     // postfix 후행감소
42     result = postfix--;
43     result.get_xy(m, n);
44     cout << "Before : " << m << ", " << n << endl;
45     postfix.get_xy(m, n);
46     cout << "After : " << m << ", " << n << endl;
47
48     return 0;
49 }
```

Before : 99, 199  
After : 99, 199  
  
Before : 100, 200  
After : 99, 199  
계속하려면 아무 키나 누르십시오 . . .

## 과제 4번

13-5. 프렌드 함수를 사용하여 coord 클래스에 관련된 - 연산자와 / 연산자를 중복하여 보아라.

```
C:\Users\Owner\Desktop\단국대학교\공학계\지향프로그래밍\C++\SelfStudy\Project1\Debug\
result.x : 8
result.y : 18

result.x : 5
result.y : 10

계속하려면 아무 키나 누르십시오 . . .

1  #include <iostream>
2  using namespace std;
3
4  class coord {
5  private:
6      int x, y;
7  public:
8      coord() { x = 0; y = 0; }
9      coord(int i, int j) { x = i; y = j; }
10     void get_xy(int &i, int &j) { i = x; j = y; }
11     friend coord operator-(coord op1, coord op2);
12     friend coord operator/(coord op1, coord op2);
13 };
14
15 // 프렌드 함수, 연산자 - 중복
16 coord operator-(coord op1, coord op2) {
17     coord temp;
18     temp.x = op1.x - op2.x;
19     temp.y = op1.y - op2.y;
20     return temp;
21 }
22
23 // 프렌드 함수, 연산자 / 중복
24 coord operator/(coord op1, coord op2) {
25     coord temp;
26     temp.x = op1.x / op2.x;
27     temp.y = op1.y / op2.y;
28     return temp;
29 }
30
31 int main() {
32     coord a(10, 20), b(2, 2), result;
33     int i, j;
34     // 연산자 - 사용
35     result = a - b;
36     result.get_xy(i, j);
37     cout << "result.x : " << i << endl;
38     cout << "result.y : " << j << '\n' << endl;
39
40     // 연산자 / 사용
41     result = a / b;
42     result.get_xy(i, j);
43     cout << "result.x : " << i << endl;
44     cout << "result.y : " << j << '\n' << endl;
45
46     return 0;
47 }
```