

Group Activity 05

(3인 혹은 4인으로 팀을 구성하여 아래의 문제를 푼다. 팀 구성은 매 시간마다 달라져도 된다.)

팀원1: _____

팀원2: _____

팀원3: _____

팀원4: _____

1. 다음 프로그램의 출력은? 컴파일 오류나 실행 오류가 나는 경우에는 이유를 간략히 설명하고, 간단한 수정으로 오류를 해결할 수 있다면 해결하라.

Program	Output
<pre>class myClass{ int pra = 5; public: int pub = 10; void set_pr(int x){ pra = x; } void set_pu(int x){ pub = x; } }; int main(){ myClass m; int a, b; a = m.pra; //LINE1 b = m.pub; //LINE2 m.set_pr(100); //LINE3 m.set_pu(200); //LINE4 return 0; }</pre>	
<pre>class Box { int capacity; Box(){} Box(double capacity){ this->capacity = capacity; } }; int main() { Box b1(10); Box b2 = Box(14); return 0; }</pre>	

<pre> class A { public: static int a; void show() { a++; cout << "a: " << a << endl; } }; int A::a = 5; int main() { A a, b; a.show(); cout << A::a << endl; b.a = 10; cout << a.a << endl; return 0; } </pre>	
<pre> class A { int a = 0; public: A() { cout<<"Object of A is created\n"; } void show() { a++; cout<<"a: "<< a << endl; } }; int main() { A a1, a2; A a3 = a1 + a2; return 0; } </pre>	
<pre> class A { int a; public: void show() { a++; cout<<"a: "<< a << endl; } void operator.() { cout << "operator . called\n"; } }; int main() { A a1, a2; return 0; } </pre>	

```

class Point {
private:
    int x, y;
public:
    Point(): x(0), y(0) { }
    Point& operator()(int dx, int dy);
    void show() {
        cout << "x = " << x << ", y = " << y;
    }
};

Point& Point::operator()(int dx, int dy)
{
    x = dx;
    y = dy;
    return *this;
}

int main() {
    Point pt;
    pt(3, 2);
    pt.show();
    return 0;
}

```

```

class complex {
    int i, j;
public:
    complex(int a, int b) {
        i = a;
        j = b;
    }

    complex operator+(complex c) {
        complex temp;
        temp.i = this->i + c.i;
        temp.j = this->j + c.j;
        return temp;
    }

    void show(){
        cout << "Complex Number: " << i
            << " + i " << j << endl;
    }
};

int main() {
    complex c1(1,2);
    complex c2(3,4);
    complex c3 = c1 + c2;
    c3.show();
    return 0;
}

```

```

class Box{
    int capacity;
public:
    Box(){}
    Box(double capacity){
        this->capacity = capacity;
    }
    bool operator<(Box b){
        return b.capacity < this->capacity ?
            true : false;
    }
};

int main() {
    Box b1(10);
    Box b2 = Box(14);
    if(b1 < b2)
        cout<<"B1's capacity is small";
    else
        cout<<"B2's capacity is small";
    return 0;
}

```

2. C++에서 overloading될 수 없는 연산자를 조사하여 나열하라,

3. 1번의 (8)번의 class complex에 +, -, *, / 연산자를 추가로 overloading하여 완성하라.