

## Programming Methodology

### Instructor: Kyuseok Shim

### Project #1: Operator overloading

**Due Date: 0:0 a.m. between 2008-4-7 & 2008-4-8**

#### Introduction

C++은 C와는 달리 다른 같은 함수 이름을 가지더라도 그 함수의 인자가 다르다면 각각의 함수를 구별한다. 예를 들어, 다음과 같은 코드가 있다고 하자.

```
int square(int x) {
    return x*x;
};

double square(double x) {
    return x*x;
};

int main() {
    cout << square(4) << endl;
    cout << square(5.5) << endl;
    return 0;
}
```

main에서 같은 square함수를 호출하지만 괄호 안의 인자가 int나 double이냐에 따라 다른 함수가 호출된다.

+, -, \*, / 등의 각종 연산자들도 함수와 같은 역할을 하지만 일반적인 함수의 형태로 사용하지 않는다. +의 함수는 실제로 operator+(...)라는 모양을 가진다. 한가지 주의할 점은 이 연산자가 특정 class안의 멤버함수로 정의된다면 인자 하나를 가지게 되고, class밖에서 정의된다면 인자 두 개를 가지게 된다. 덧셈을 하려면 a + b와 같이 a, b 두 개의 인자가 필요하지만 멤버함수일 경우 첫번째 인자는 자기 자신이 되기 때문이다. 'a+b;'는 'a.operator+(b);'의 모양으로 실행된다고 생각할 수 있다. 주어진 연산이 float형태의 변수의 덧셈이라면 그에 맞는 덧셈(실제덧셈)을 하게 되고, string형태의 변수의 덧셈이라면 string 타입의 덧셈(concatenate)을 하게 된다. 만약, 컴파일러에 기본적으로 내장되어 있지 않은 덧셈이라면 어떻게 더하는 것인지 정하지 않았다는 내용의 에러가 발생할 것이다. 프로그래머 자신이 만든 class에서 특정연산자를 사용하고자 할 때는 그 연산자가 어떻게 동작할 것인지에 대해 미리 컴퓨터에게 알려주어야 한다. 그것이 operator overloading(연산자

오버로딩)이다.

## Implementation

### 1. class Complex 구현

- Constructor ( $0+0i$ 를 만드는 default constructor 포함)
- +, \* operator overloading
- >>, << operator overloading (형식지정)

### 2. class Quadratic 구현

- Constructor (0를 만드는 default constructor 포함)
- + operator overloading
- >>, << operator overloading (형식지정)
- eval(...) function (주어진 2차식에 x값을 대입하면 결과를 출력해주는 함수)
- eq() function ('주어진 2차식=0'의 방정식의 해를 구하는 함수)

## Input

키보드를 통해서 입력을 할 경우 숫자를 하나씩 입력하도록 한다. 예를 들어 5-9이라는 복소수를 입력할 때에는 " 5 -9 "를 입력한다. " $5x^2+72x-100$ "을 입력할 때에는 " 5 72 -100"을 입력한다.

입력은 double을 받는다. 즉, " $0.5 + 54.23i$ "와 같은 값을 입력할 수도 있다.

## Output

Complex 타입을 cout<<을 이용하여 출력할 경우에는 "(실수부)빈칸+빈칸(허수부)i"로 출력한다. 즉 Complex aaa에  $5+7i$ 가 저장되어 있다면 "cout<<aaa<<endl;"을 실행하면 "(5) + (7)i"가 출력 되어야 한다. 만약 실수부나 허수부의 값이 음수라면 "(-5) + (7)i" 혹은 "(5) + (-7)i" 등으로 출력되어야 한다. 실수부와 +, +과 허수부 사이에는 반드시 빈칸을 하나 삽입한다

Quadratic 타입 또한 cout<<을 이용하여 출력할 경우 "(계수)x<sup>2</sup> + (계수)x + (상수)"로 출력한다. Quadratic bbb에  $5x^2+72x-100$ 가 저장되어 있다면 "cout<<bbb<<endl;"는 "(5)x<sup>2</sup> + (72)x + (-100)"을 출력해야 한다. Quadratic 또한 항과 +, +과 항 사이에는 반드시 빈칸을 하나 삽입한다.

최종적으로 출력은 두개의 복소수를 입력받아서 그 두수의 합과 곱을 출력하고, 두개의 2차식을 입력받아서 두식의 합과 첫번째 식의 x에 특정복소수를 입력했을 경우의 식의 값과 두번째 식을 우변이 0인 방정식으로 만들었을 때의 해를 출력한다. 첫번째 식에 특정값을 대입하였을 때의 값은 그 값만을 출력하고, 방정식의 해는 "(방정식 출력) ==> x = s1 or x = s2"의 형식으로 출력한다.

```

class Complex {
    private:
        double real;
        double image;
    public:
        Complex();
        Complex(double r, double i);
        Complex operator+(Complex second); // Complex + Complex
        Complex operator+(double second); // Complex + double
        Complex operator*(Complex second); // Complex * Complex
        Complex operator*(double second); // Complex * double

        friend ostream& operator<<(ostream& output, Complex c); // 출력
        friend istream& operator>>(istream& output, Complex c); // 입력
};

class Quadratic {
    private:
        double a; // ax^2 + bx + c
        double b;
        double c;
    public:
        Quadratic();
        Quadratic(double x, double y, double z);
        double eval(double x);
        void eq();
        Quadratic operator+(Quadratic second); // polynomial+polynomial

        friend ostream& operator<<(ostream& output, Quadratic q); // 출력
};

// 위의 함수들의 body를 구현
// 주어진 함수 이외의 다른 함수를 정의하여 사용해도 무방. 예를 들어 set() 등의 함수
// main함수에서 호출되는 함수는 꼭 주어진 함수여야 함

```

```

int main() {
    double x,y,z;
    cout << "complex: ";
    cin >> x >> y;
    Complex c1(x,y);
    Complex c2;
    cout << "complex: ";
    cin >> c2;
    cout << "(" << c1 << ") + (" << c2 << ") = " << c1+c2 << endl;
    cout << "(" << c1 << ") * (" << c2 << ") = " << c1*c2 << endl;

    cout << "polynomial: ";
    cin >> x >> y >> z;
    Quadratic q1(x,y,z);
    cout << "polynomial: ";
    cin >> x >> y >> z;
    Quadratic q2(x,y,z);
    cout << "(" << q1 << " + " << q2 << ") = ";
    cout << q1+q2 << endl;
    cout << "x : ";
    cin >> x;
    cout << q1.eval(x) << endl;
    q2.eq();

    system("PAUSE");
    return EXIT_SUCCESS;
}

```

## 프로그램 실행 예시

```
> project1.exe
complex: 1(키보드입력) 2(키보드입력)
complex: 5(키보드입력) 7(키보드입력)
((1) + (2)i) + ((5) + (7)i) = (6) + (9)i
((1) + (2)i) * ((5) + (7)i) = (-9) + (17)i
polynomial: 1(키보드입력) -1(키보드입력) 3(키보드입력)
polynomial: 1(키보드입력) 5(키보드입력) 6(키보드입력)
((1)x^2 + (-1)x + (3)) + ((1)x^2 + (5)x + (6)) = (2)x^2 + (4)x + (9)
x : 1(키보드입력)
3
(1)x^2 + (5)x + (6) = 0 ==> x = (2) + (0)i or x = (3) + (0)i
```

## 제출사항

### 1. 제출 방법 및 내용

#### 1) 프로젝트 구현 파일 제출 (DEV C++을 사용하세요!!!)

- 파일은 **ee** 홈페이지에 과제 제출을 이용하여 제출한다.
- 소스 파일과 컴파일을 할 수 있는 파일과 실행파일을 압축하여 제출한다.
- 각 제출 파일 이름은 다음의 형식을 따른다.

pro\_1\_학번.zip

예) pro\_1\_2007-12345.zip

- 각 소스 파일 상단에 **E-mail**주소와 작성자 학번 이름을 적는다.

#### 2) 보고서 제출

- 간단한 설명과 **discussion**을 하드카피로 수업시간 전에 제출한다.

### 2. 제출 일시

파일 제출 마감일: **2008년 4월 7일과 8일 사이 새벽 0:00**

구현 파일 제출 시각은 홈페이지 과제 제출 시각에 준한다.

보고서는 **2008년 4월 8일 수업 시작 전에 제출 (강의실에서)**

### 채점 규칙

몇 가지 값들을 입력하여 각각의 함수들이 정확히 동작하는지 확인하여 채점

딜레이는 하루에 10%씩 감점

Copy 발견 시 모든 숙제 0점 처리