



🏠 / C++프로그래밍과실습 (CB3500572-062) / 실습 072 - + 연산자 오버로딩

개요

제출

편집

코딩 결과

## 실습 072 - + 연산자 오버로딩

제출 마감일: 2023-05-04 23:59

업로드 가능한 파일 수: 2

제출 방식: 개인

### 목적

이 실습은 class 의 + / - 연산자 오버로딩을 연습합니다.

### 설명

복소수를 다루는 클래스를 정의해 봅시다.

복소수는 실수부와 허수부로 구성되어 있으므로 다음과 같이 클래스의 멤버 변수를 선언할 수 있습니다.

```
class Complex {
public:
    Complex(double real=0, double imaginary=0) : r(real), i(imaginary) {};

    // operator << 연산자 오버로딩 필요함 - free 함수

    // operator + 연산자 오버로딩 필요함 - 멤버 함수

private:
    double r{0}, i{0};
};
```

두 복소수를 + 연산자를 이용해서 덧셈을 수행하도록 하고 싶습니다.

예를 들어, int a = 1, b=1; int c = a + b; 와 같이

Complex c1(1, 1); c2(2, 2);

C++ 프로그래밍과 실습

Complex c3 = c1 + c2, 와 같습니다.

이를 위해 Complex 멤버 함수로 operator + 함수를 정의합니다.

```
Complex operator + (const Complex& rhs);
```

operator - 함수도 정의해 보겠습니다.

```
Complex operator - (const Complex& rhs);
```

Complex 클래스를 사용해 보겠습니다.

```
Complex c1 (3, 2);
```

```
Complex c2 (1, 1);
```

```
cout << c1 + c2 << '\n';
```

```
cout << c1 - c2 << '\n';
```

기대되는 출력은 4+3i 와 2+2i 입니다.

자신이 만든 새로운 타입인 복소수 Complex 클래스가 마치 int 타입처럼 동작되는 것을 확인할 수 있습니다.

C++ 은근 매력 있지 않습니까?

문제

Complex 클래스에서 + 연산자와 - 연산자를 정의하시오.

입력

없음

출력

4+3i

2+1i

3+2i

1+1i

## 제출파일

Complex.cpp

72.csv

## 참고

Complex.h -----

```
#include <ostream>

class Complex {
public:
    Complex(double real=0, double imaginary=0) : r(real), i(imaginary) {};
    Complex operator + (const Complex rhs);
    Complex operator - (const Complex rhs);
    friend std::ostream& operator << (std::ostream& os, const Complex& c);
private:
    double r{0}, i{0};
};

std::ostream& operator << (std::ostream& os, const Complex& c);
```

ComplexAddTest.cpp -----

```
int main() {
    Complex c1(3, 2);
    Complex c2(1, 1);

    std::cout << c1 + c2 << std::endl;
    std::cout << c1 - c2 << std::endl;

    std::cout << c1 << std::endl;
    std::cout << c2 << std::endl;
}
```