



🏠 / C++프로그래밍과실습 (CB3500572-062) / 실습 071 출력 연산자

개요

제출

편집

코딩 결과

실습 071 출력 연산자

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

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

표준 출력 (cout) 으로 손쉽게 복소수의 정보를 출력하고 싶습니다.

free function 으로 정의해 보겠습니다. (이유는 강의영상 참고)

```
ostream& operator << (ostream& os, const complex& c) {
```

```
    // ...
```

```
//your code nere
```

```
}
```

단, 이때 operator << () 함수 안에서 complex 클래스의 private 멤버 변수 r 과 i 에 접근할 수 있어야 하므로, complex 클래스 내부에서 free function 을 friend 로 해 줘야 합니다.

complex 클래스를 사용해 보겠습니다.

```
complex c1 (3, 2);
complex c2 (1, 1);

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

기대되는 출력은 3+2i, 1+1i 입니다.

문제

complex 클래스에서 << 연산자를 정의하시오.

<참고>

```
int main() {
    Complex c1(3, 2), c2(1, 1);
    Complex c3(c1);
    std::cout << c1 << std::endl;
    std::cout << c2 << std::endl;
    std::cout << c3 << std::endl;
}
```

입력

없음

출력

3+2i

1+1i

3+2i

제출파일

Complex.cpp

71.csv

참고

Complex.h-----

```
#include <ostream>
```

```
class Complex {
```

```
public:
```

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

```
    // operator << 를 구현하시오 (free 함수로 구현)
```

```
    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);
```

ComplexTest.cpp -----

```
int main() {
```

```
    Complex c1(3, 2), c2(1, 1);
```

```
    Complex c3(c1);
```

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

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

```
    std::cout << c3 << std::endl;
```

```
}
```