

Section 1A. Class Cmplx (30점)

1A.1 class Cmplx 정의

```

/** Class_Cmplx.h */

#ifndef CMPLX_H
#define CMPLX_H
#include <iostream>

using namespace std;

class Cmplx
{
public:
    Cmplx(double real=0.0, double imag=0.0); // constructor
    void inputCmplx();
    void printCmplx();
    const Cmplx addCmplx(const Cmplx&);
    const Cmplx subCmplx(const Cmplx&);
    const Cmplx mulCmplx(const Cmplx&);
    const Cmplx divCmplx(const Cmplx&);

private:
    double real;
    double imag;
};

#endif

```

- class Cmplx는 데이터 멤버로 복소수의 실수부인 real과 허수부인 imag을 double 자료형으로 포함
- class Cmplx는 Class_Cmplx.h 헤더 파일에 구현

1A.2 class Cmplx 멤버함수 구현

- class Cmplx의 생성자는 real과 imag 값을 인수로 전달받아 각각 복소수의 실수부와 허수부 값으로 초기화
- inputCmplx() 함수는 복소수의 실수부와 허수부 값을 표준입력장치(키보드)로부터 소수점 이하 값을 포함하는 실수 자료형으로 2개 입력받아, 각각 real과 imag 값으로 설정
- printCmplx() 함수는 class Cmplx 변수의 값을 소괄호안에 실수부 및 허수부, j가 포함된 "(123.45 +/- 567.89j)" 형식으로 출력. 실수부와 허수부의 double 자료형 값은 소수점이하 2자리까지 표현하며, 허수부 값의 +/-에 따른 부호(sign)를 출력하고, 허수부의 절대값을 출력하며, 허수부 다음에 'j'를 출력하여 복소수로 표현함. 허수부의 부호(sign) 앞뒤에 공란을 둘 것.
- class Cmplx의 addCmplx(), subCmplx(), mulCmplx(), divCmplx() 함수는 인수로 복소수 객체를 call-by-reference 방식으로 전달받으며, 각각 복소수의 덧셈, 뺄셈, 곱셈 및 나눗셈 연산을 수행하며, 그 결과를 class Cmplx 객체로 반환함. call-by-reference로 전달받은 인수들은 const로 보호하며, 반환되는 복소수 객체도 const로 보호할 것.
- class Cmplx의 멤버함수들은 Class_Cmplx.cpp 파일에 구현

1A.3 main() 함수 구현

```
/** main.cpp */

#include <iostream>
#include "Class_Cmplx.h"

using namespace std;

void main()
{
    Cmplx c1, c2, c3, c4, c5, c6;

    c1.inputCmplx();
    cout << " c1 = ";    c1.printCmplx(); cout << endl;

    c2.inputCmplx();
    cout << " c2 = ";    c2.printCmplx(); cout << endl;

    c3 = c1.addCmplx(c2);
    cout << " c3 = c1 + c2 = "; c3.printCmplx(); cout << endl;

    c4 = c1.subCmplx(c2);
    cout << " c4 = c1 - c2 = "; c4.printCmplx(); cout << endl;

    c5 = c1.mulCmplx(c2);
    cout << " c5 = c1 * c2 = "; c5.printCmplx(); cout << endl;

    c6 = c1.divCmplx(c2);
    cout << " c6 = c1 / c2 = "; c6.printCmplx(); cout << endl;
}
```

1A.4 실행 결과 화면출력

```
-
input cmplx.real and cmplx.imag : 5.67 7.89
c1 = (5.67 + 7.89j)
input cmplx.real and cmplx.imag : 1.23 4.56
c2 = (1.23 + 4.56j)
c3 = c1 + c2 = (6.90 + 12.45j)
c4 = c1 - c2 = (4.44 + 3.33j)
c5 = c1 * c2 = (-29.00 + 35.56j)
c6 = c1 / c2 = (1.93 - 0.72j)
```