#include <iostream>

#include <string>

#include <assert.h>

/\*

\* A class representing complex number

\* It has both real and imaginary part

\*/

class ComplexNumber

{

int real;

int imaginary;

public:

ComplexNumber() :

real(0), imaginary(0)

{}

ComplexNumber(int r, int i) :

real(r), imaginary(i)

{}

void print()

{

int img = imaginary < 0 ? -imaginary : imaginary;

std::cout << real << (imaginary < 0 ? " - ": " + ") << "i" << img << std::endl;

}

// Overloaded unary minus operator

ComplexNumber operator-() const;

};

/\*

\* Overloaded unary minus operator as member function.

\* It returns a new Object.

\*/

ComplexNumber ComplexNumber::operator-() const

{

return ComplexNumber(-(this->real), -(this->imaginary) );

}

int main()

{

// Create a Complex Number Object

ComplexNumber c1(2, -3);

std::cout<<"c1 = ";

c1.print();

// Call the unary operator minus on c1 and

// store the returned in a new object

ComplexNumber c2 = -c1;

std::cout<<"c2 = ";

c2.print();

return 0;

}

Output :

c1 = 2 - i3

c2 = -2 + i3