

## EE 154 Laboratory 8

### Question:

Write and study the following C++ program carefully:

```
x: (0, 0)
y: (4.3, 8.2)
z: (3.3, 1.1)

x = y + z:
(7.6, 9.3) = (4.3, 8.2) + (3.3, 1.1)

x = y - z:
(1, 7.1) = (4.3, 8.2) - (3.3, 1.1)
```

### HEADER FILE :

(Save under “Header Files” as a header file with “Complex.h” name)

```
// Complex class definition.
#ifndef COMPLEX_H
#define COMPLEX_H

class Complex
{
public:
    Complex( double = 0.0, double = 0.0 ); // constructor
    Complex operator+( const Complex & ) const; // addition
    Complex operator-( const Complex & ) const; // subtraction
    void print() const; // output
private:
    double real; // real part
    double imaginary; // imaginary part
}; // end class Complex

#endif
```

### SOURCE FILE:

(Save under “Source Files” as a source file with “Complex.cpp” name)

```
// Complex class member-function definitions.
#include <iostream>
using std::cout;

#include "Complex.h" // Complex class definition

// Constructor
Complex::Complex( double realPart, double imaginaryPart )
    : real( realPart ),
      imaginary( imaginaryPart )
{
    // empty body
} // end Complex constructor

// addition operator
Complex Complex::operator+( const Complex &operand2 ) const
```

```

{
    return Complex( real + operand2.real,
        imaginary + operand2.imaginary );
} // end function operator+

// subtraction operator
Complex Complex::operator-( const Complex &operand2 ) const
{
    return Complex( real - operand2.real,
        imaginary - operand2.imaginary );
} // end function operator-

// display a Complex object in the form: (a, b)
void Complex::print() const
{
    cout << '(' << real << ", " << imaginary << ')';
} // end function print

```

### SOURCE FILE:

(Save under “Source Files” as a source file with “Solution.cpp” name)

```

// Complex class test program.
#include <iostream>
using std::cout;
using std::endl;

#include "Complex.h"

int main()
{
    Complex x;
    Complex y( 4.3, 8.2 );
    Complex z( 3.3, 1.1 );

    cout << "x: ";
    x.print();
    cout << "\ny: ";
    y.print();
    cout << "\nz: ";
    z.print();

    x = y + z;
    cout << "\n\nx = y + z:" << endl;
    x.print();
    cout << " = ";
    y.print();
    cout << " + ";
    z.print();

    x = y - z;
    cout << "\n\nx = y - z:" << endl;
    x.print();
    cout << " = ";
    y.print();
    cout << " - ";
    z.print();
    cout << endl;
    getchar();
    return 0;
} // end main

```