```cpp
/*
280
Complex 1
Elliot Shaw
*/

#include <iostream>
using namespace std;

class Complex {
private:
    double rPart, iPart;
public:
    //constructors
    Complex();
    Complex(double, double);

    //getters/accessors
    //double getReal();
    double getReal();
    //double getImag();
    double getImag();

    //setters/mutators
    void set(double, double);
    //void  setReal(double);
    void setReal(double);
    //void setImag(double);
    void setImag(double);

    //utility
    //bool equal(Complex);
    //friend bool equal(Complex, Complex);
    void display();
    void load();

    //arithmetic
    void add(Complex);
    void sub(Complex);
    void mult(Complex);
    friend Complex add(const Complex, const Complex);
    friend Complex sub(const Complex, const Complex);
    friend Complex mult(const Complex, const Complex);
    friend Complex operator + (const Complex, const Complex);
    friend Complex operator - (const Complex, const Complex);
    friend Complex operator * (const Complex, const Complex);
    friend ostream& operator << (ostream&, const Complex&);
    friend istream& operator >> (istream&, Complex&);
};
```

```
50
51  Complex::Complex() {
52      rPart = iPart = 0;
53  } //Complex
54
55  Complex::Complex(double rVal, double iVal) {
56      rPart = rVal;  iPart = iVal;
57  } //Complex
58
59  //Accessors
60  double Complex::getReal() {
61      return rPart;
62  }//getReal
63
64  double Complex::getImag() {
65      return iPart;
66  }//getImag
67
68  //setters
69  void Complex::set(double rVal, double iVal) {
70      rPart = rVal;  iPart = iVal;
71  } //set
72
73  void Complex::setReal(double newR) {
74      rPart = newR;
75  }//setReal
76
77  void Complex::setImag(double newI) {
78      iPart = newI;
79  }//setImag
80
81
82  //utility
83  void Complex::load() {
84      cout << "Enter rp and ip: ";
85      cin >> rPart >> iPart;
86  } //load
87
88  void Complex::display() {
89      cout << rPart;
90      if (iPart >= 0) cout << " + " << iPart;
91      else cout << " - " << -iPart;
92      cout << "i" << endl;
93  } //display
94
95
96  //math
97  void Complex::add(const Complex val) {
98      rPart += val.rPart;
```

```
 99      iPart += val.iPart;
100  } //add
101
102  Complex add(const Complex v1, const Complex v2) {
103      Complex t = Complex();
104      t.rPart = v1.rPart + v2.rPart;
105      t.iPart = v1.iPart + v2.iPart;
106      return t;
107  } //add friend
108
109  Complex operator + (const Complex v1, const Complex v2) {
110      return add(v1, v2);
111  } //+
112
113  void Complex::sub(const Complex val) {
114      rPart -= val.rPart;
115      iPart -= val.iPart;
116  } //sub
117
118  Complex sub(const Complex v1, const Complex v2) {
119      Complex t = Complex();
120      t.rPart = v1.rPart - v2.rPart;
121      t.iPart = v1.iPart - v2.iPart;
122      return t;
123  } //sub friend
124
125  Complex operator - (const Complex v1, const Complex v2) {
126      return sub(v1, v2);
127  } //-
128
129  void Complex::mult(const Complex val) {
130      double finalR, finalI;
131      finalR = (rPart * val.rPart) + (iPart * val.iPart);
132      finalI = (rPart * val.iPart) + (iPart * val.rPart);
133      rPart = finalR;
134      iPart = finalI;
135  }//mult
136
137  Complex mult(const Complex v1, const Complex v2) {
138      Complex t = Complex();
139      t.rPart = (v1.rPart * v2.rPart) + (v1.iPart * v2.iPart);
140      t.iPart = (v1.rPart * v2.iPart) + (v1.iPart * v2.rPart);
141      return t;
142  }//mult friend
143
144  Complex operator * (const Complex v1, const Complex v2) {
145      return mult(v1, v2);
146  }//*
147
```

```
148 ostream& operator << (ostream& out, const Complex& v)
149 {
150     out << v.rPart;
151     if (v.iPart >= 0) out << " + " << v.iPart;
152     else out << " - " << -v.iPart;
153     out << 'i';
154     return out;
155 }//<<
156 
157 istream& operator >> (istream& in, Complex& v)
158 {
159     cout << "Enter rPart ";
160     in >> v.rPart;
161     cout << "Enter iPart ";
162     in >> v.iPart;
163     return in;
164 }//>>
165 
166 int main() {
167     Complex c1 = Complex(1, -1);
168     Complex c2 = Complex(3, -4);
169 
170     c1.add(c2);
171     cout << c1 << endl;
172     c1 = c1 - c2;
173     cout << c1 << endl;
174 
175     c1.sub(c2);
176     cout << c1 << endl;
177     c1 = c1 + c2;
178     cout << c1 << endl;
179 
180     Complex c3 = add(c1, c2);
181     cout << c3 << endl;
182 
183     c3 = sub(c1, c2);
184     cout << c3 << endl;
185 
186     c3 = mult(c1, c2);
187     cout << c3 << endl;
188 
189     c1 = c1 * c2;
190     cout << c1 << endl;
191 
192     Complex c4;
193     cin >> c4;
194 
195     cout << c4;
196 } //main
```