

```
1 // Justin Dang Student ID: 1148267
2 /*
3  Instantiates and prints two complex numbers
4
5  checks if the complex numbers are equal to each other
6
7  Calculates, stores, and prints the sum, difference, and product of the two      ↗
   numbers
8
9  Calculates and prints the cubes of both complex numbers
10 */
11
12 #include <iostream>
13 using namespace std;
14
15 class CNumber {
16     friend ostream& operator << (ostream&, const CNumber&);
17     friend istream& operator >> (istream&, CNumber&);
18 public:
19     CNumber(float = 0.0, float = 0.0);
20
21     CNumber operator + (const CNumber& rhs) const {
22         return CNumber(num1 + rhs.num1, num2 + rhs.num2);
23     }
24     CNumber operator - (const CNumber& rhs) const {
25         return CNumber(num1 - rhs.num1, num2 - rhs.num2);
26     }
27     CNumber operator * (const CNumber& rhs) const {
28         return CNumber(((num1 * rhs.num1) - (num2 * rhs.num2)), ((num1 *      ↗
           rhs.num2) + (num2 * rhs.num1)));
29     }
30     bool operator == (const CNumber& rhs) const {
31         return (num1 == rhs.num1 && num2 == rhs.num2);
32     }
33     bool operator != (const CNumber& rhs) const {
34         return (num1 != rhs.num1 || num2 != rhs.num2);
35     }
36     CNumber& operator ++ () {
37         CNumber temp = *this; // 1
38         num1 = (temp.num1 * temp.num1 * temp.num1) - ((float)3 * temp.num1 *      ↗
           (temp.num2 * temp.num2)); // 2
39         num2 = ((float)3 * (temp.num1 * temp.num1) * temp.num2) - (temp.num2 *      ↗
           temp.num2 * temp.num2); // 3
40
41         return *this; // 4
42     }
43     CNumber operator ++ (int) {
44         CNumber temp = *this;
45
46         num1 = (temp.num1 * temp.num1 * temp.num1) - ((float)3 * temp.num1 *      ↗
           (temp.num2 * temp.num2));
47         num2 = ((float)3 * (temp.num1 * temp.num1) * temp.num2) - (temp.num2 *      ↗
```

```
        temp.num2 * temp.num2);
48
49     return temp;
50 }
51 void SetNum1(float);
52 void SetNum2(float);
53 private:
54     float num1, num2;
55 };
56 CNumber::CNumber(float a, float b) {
57     SetNum1(a);
58     SetNum2(b);
59 }
60 void CNumber::SetNum1(float a) {
61     num1 = a;
62 }
63 void CNumber::SetNum2(float a) {
64     num2 = a;
65 }
66 ostream& operator << (ostream& output, const CNumber& cNum) {
67     if (cNum.num2 >= 0) {
68         output << cNum.num1 << '+' << cNum.num2 << 'i';
69     }
70     else if (cNum.num2 <= 0) {
71         output << cNum.num1 << cNum.num2 << 'i';
72     }
73     return output;
74 }
75 istream& operator >> (istream& input, CNumber& cNum) {
76     input >> cNum.num1 >> cNum.num2;
77     return input;
78 }
79
80 int main()
81 {
82     CNumber num1, num2(10.0, 5.0), num3;
83
84
85
86     cout << "1st complex number: " << num1 << endl;
87     cout << "2nd complex number: " << num2 << "\n\n";
88
89     if (num1 == num2)
90         cout << num1 << " is equal to " << num2 << "\n\n";
91     else if (num1 != num2)
92         cout << num1 << " is not equal to " << num2 << "\n\n";
93
94     num3 = num1 + num2;
95     cout << "The sum of the complex numbers is: " << num3 << "\n\n";
96
97     num3 = num1 - num2;
98     cout << "The difference of the complex numbers is: " << num3 << "\n\n";
```

```
99
100     num3 = num1 * num2;
101     cout << "The product of the complex numbers is: " << num3 << "\n\n";
102
103
104
105     num1++;
106     cout << "The cube of num 1 is: " << num1 << endl;
107
108     num2++;
109     cout << "The cube of num 2 is: " << num2 << endl;
110     return 0;
111 }
112 /*
113 1st complex number: 0+0i
114 2nd complex number: 10+5i
115
116 0+0i is not equal to 10+5i
117
118 The sum of the complex numbers is: 10+5i
119
120 The difference of the complex numbers is: -10-5i
121
122 The product of the complex numbers is: 0+0i
123
124 The cube of num 1 is: 0+0i
125 The cube of num 2 is: 250+1375i
126 */
127
```