

Steven Nguyen

Jake Nugent

CIS 368

Assignment 5

OUTPUT:

```
PS C:\gethub\objectOrientedDesignAndProrams\object-oriented-design\assignFive> java .\assignmentF.java
Enter the first complex number: 3.5 5.5
Enter the second complex number: -3.5 1
(3.5 + 5.5i) + (-3.5 + 1.0i) = 0.0 + 6.5i
(3.5 + 5.5i) - (-3.5 + 1.0i) = 7.0 + 4.5i
(3.5 + 5.5i) * (-3.5 + 1.0i) = -17.75 + -15.75i
(3.5 + 5.5i) / (-3.5 + 1.0i) = -0.5094339622641509 + -1.7169811320754718i
|3.5 + 5.5i| = 6.519202405202649
false
3.5
5.5
C1 is not the same as C2
```

COMPLEX IMPLEMENT:

```

51 class Complex implements Cloneable, Comparable<Complex> {
52     private double a;
53     private double b;
54
55     public Complex() {
56
57     }
58
59     public Complex(double a_in) {
60         this.a = a_in;
61         this.b = 0;
62     }
63
64     Complex(double a_in, double b_in) {
65         this.a = a_in;
66         this.b = b_in;
67     }
68
69
70     public double getRealPart() {
71         return a;
72     }
73
74     public double getImaginaryPart() {
75         return b;
76     }
77
78     public Complex add(Complex secondComplex) {
79         double newA = a + secondComplex.a;
80         double newB = b + secondComplex.b;
81         return new Complex(newA, newB);
82     }
83
84     public Complex subtract(Complex secondComplex) {
85         // stilll needs to be implemented
86         double newA = a - secondComplex.a;
87         double newB = b - secondComplex.b;
88         return new Complex(newA, newB);
89     }
90
91     public Complex multiply(Complex secondComplex) {
92         double newA = a * secondComplex.a - b * secondComplex.b;
93         double newB = b * secondComplex.a + a * secondComplex.b;
94         return new Complex(newA, newB);
95     }
96

```

## SELF IMPLEMENTED FUNCTIONS:

```
97     public Complex divide(Complex secondComplex) {
98         double newA = (a * secondComplex.a + b * secondComplex.b)
99         / ((secondComplex.a * secondComplex.a) + (secondComplex.b * secondComplex.b));
100     double newB = (b * secondComplex.a - a * secondComplex.b)
101     / ((secondComplex.a * secondComplex.a) + (secondComplex.b * secondComplex.b));
102     return new Complex(newA, newB);
103 }
104 public double abs() {
105     return Math.sqrt(a * a + b * b);
106 }
107 @Override
108 public String toString() {
109     if (b != 0) {
110         return a + " + " + b + "i";
111     }
112     return a + "";
113 }
114 @Override
115 public int compareTo(Complex c) {
116     if ((c.a == this.a) && (c.b == this.b))
117         return 1;
118     else {
119         return 0;
120     }
121 }
122 @Override
123 public Complex clone() {
124     return new Complex(this.a, this.b);
125 }
126 }
127
128
129
```