

Review: Chapter 3 Programming

3.18 Classes

```
Assignment_03_18 > J Assignment_03_18_Combolock.java > ...
1  public class Assignment_03_18_Combolock {
2
3      private Assignment_03_18_Combination combo;
4
5      /**
6       * This class represents a 3 digit combination lock, initilized to 0,0,0
7       * @author Weston Shakespear
8       */
9      public Assignment_03_18_Combolock() {
10         this.combo = new Assignment_03_18_Combination(_x:0, _y:0, _z:0);
11     }
12
13     /**
14      * This class represents a 3 digit combination lock
15      * @author Weston Shakespear
16      * @param _num1 First digit
17      * @param _num2 Second digit
18      * @param _num3 Third digit
19      */
20     public Assignment_03_18_Combolock(int _num1, int _num2, int _num3) {
21         this.combo = new Assignment_03_18_Combination(_num1, _num2, _num3);
22     }
23
24     /**
25      * This method tries to open the combination lock
26      * @author Weston Shakespear
27      * @param _num1 First test digit
28      * @param _num2 Second test digit
29      * @param _num3 Third test digit
30      * @return boolean Unlock status
31      */
32     public boolean open(int _num1, int _num2, int _num3) {
33         return this.combo.eval(_num1, _num2, _num3);
34     }
35
36     /**
37      * This method changes the code, as long as the lock is unlocked
38      * @author Weston Shakespear
39      * @param _num1 First test digit
40      * @param _num2 Second test digit
41      * @param _num3 Third test digit
42      * @param _newNum1 First new digit
43      * @param _newNum2 Second new digit
44      * @param _newNum3 Third new digit
45      * @return boolean Code changed status
46      */
47     public boolean changeCombo(int _num1, int _num2, int _num3, int _newNum1, int _newNum2, int _newNum3) {
48
49         if (this.open(_num1, _num2, _num3))
50         {
51             this.combo = new Assignment_03_18_Combination(_newNum1, _newNum2, _newNum3);
52             return true;
53         }
54         return false;
55     }
56
57 }
```

Review: Chapter 3 Programming

```
Assignment_03_18 > J Assignment_03_18_Combination.java > ...
1  public class Assignment_03_18_Combination {
2
3      private int x;
4      private int y;
5      private int z;
6
7      /**
8       * This class represents a simple 3 digit code
9       * @author Weston Shakespear
10      * @param _x First digit
11      * @param _y Second digit
12      * @param _z Third digit
13      */
14      public Assignment_03_18_Combination(int _x, int _y, int _z)
15      {
16          this.x = _x;
17          this.y = _y;
18          this.z = _z;
19      }
20
21      /**
22       * Check to see if the provided code matches
23       * @author Weston Shakespear
24       * @param _x First digit
25       * @param _y Second digit
26       * @param _z Third digit
27       * @return boolean Whether or not the code matches
28       */
29      public boolean eval(int _x, int _y, int _z)
30      {
31          return (this.x == _x && this.y == _y && this.z == _z);
32      }
33
34  }
```

3.18 Testing Output

```
c:\Users\wes\github-repos\cs2420_summer2023\Chapter3\Assignment_03_18 - VS Code Console
Starting tests
Ending tests: No red output in eclipse means success
Press any key to continue . . . _
```

Review: Chapter 3 Programming

3.31 Constructors

```
8
9  /**
10   * This class provides operations for complex numbers, initially set at 0+0i
11   * @author Weston Shakespear
12   */
13  public Assignment_03_31_BigComplex() {
14      this.real = new BigDecimal(val:0);
15      this.imaginary = new BigDecimal(val:0);
16  }
17
18  /**
19   * This class provides operations for complex numbers, with an initial complex part of 0i
20   * @author Weston Shakespear
21   * @param _real The real portion of the number
22   */
23  public Assignment_03_31_BigComplex(BigDecimal _real) {
24      this.real = _real;
25      this.imaginary = new BigDecimal(val:0);
26  }
27
28  /**
29   * This class provides operations for complex numbers
30   * @author Weston Shakespear
31   * @param _real The real portion of the number
32   * @param _imaginary The complex portion of the number
33   */
34  public Assignment_03_31_BigComplex(BigDecimal _real, BigDecimal _imaginary) {
35      this.real = _real;
36      this.imaginary = _imaginary;
37  }
38
39  /**
40   * This class provides operations for complex numbers, initilized from a string
41   * @author Weston Shakespear
42   * @param s The complex number in the form of "13 + 7i"
43   */
44  public Assignment_03_31_BigComplex(String s) {
45      // Split the real and complex string parts out
46      String[] parts = s.split(regex:" \\+ ");
47
48      if (parts.length == 1)
49      {
50          // All we have is the real part
51          real = new BigDecimal(s);
52          imaginary = new BigDecimal(val:0);
53      } else {
54          // There is both a real and a complex part
55          real = new BigDecimal(parts[0]);
56          // Cleanup complex part
57          String imag = parts[1].replace(target:"i", replacement:"");
58          imaginary = new BigDecimal(imag);
59      }
60  }
```

Review: Chapter 3 Programming

3.31 Arithmetic Methods

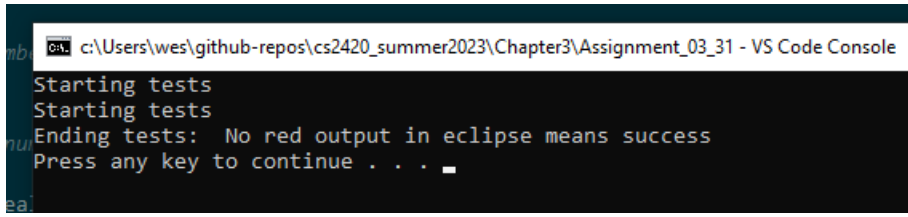
```
62  /**
63   * This method allows adding two complex numbers by adding the parts
64   * @author Weston Shakespear
65   * @param a The first complex number
66   * @param b The second complex number
67   * @return BigComplex The result
68   */
69  public static Assignment_03_31_BigComplex add(Assignment_03_31_BigComplex a, Assignment_03_31_BigComplex b) {
70      BigDecimal real = a.real;
71      real = real.add(b.real);
72
73      BigDecimal imag = a.imaginary;
74      imag = imag.add(b.imaginary);
75
76      return new Assignment_03_31_BigComplex(real, imag);
77  }
78
79  /**
80   * This method allows subtracting two complex numbers by subtracting the parts
81   * @author Weston Shakespear
82   * @param a The first complex number
83   * @param b The second complex number
84   * @return BigComplex The result
85   */
86  public static Assignment_03_31_BigComplex subtract(Assignment_03_31_BigComplex a, Assignment_03_31_BigComplex b) {
87      BigDecimal real = a.real;
88      real = real.subtract(b.real);
89
90      BigDecimal imag = a.imaginary;
91      imag = imag.subtract(b.imaginary);
92
93      return new Assignment_03_31_BigComplex(real, imag);
94  }
95
96  /**
97   * Returns true if the two complex numbers are equal
98   * @author Weston Shakespear
99   * @param a An instance of the BigComplex class
100  * @return boolean Whether or not the value is equal to this
101  */
102  public boolean equals(Assignment_03_31_BigComplex a) {
103      return (this.real.equals(a.real)) && (this.imaginary.equals(a.imaginary));
104  }
105
106  /**
107   * Returns true if the two complex numbers are equal
108   * @author Weston Shakespear
109   * @param a An instance of the BigComplex class
110   * @return boolean Whether or not the value is equal to this
111   */
112  @Override
113  public boolean equals(Object a) {
114      if (!(a instanceof Assignment_03_31_BigComplex))
115      {
116          return false;
117      }
118      return equals((Assignment_03_31_BigComplex) a);
119  }
```

3.31 toString Method

```
121  /**
122   * Returns a string representation of the complex number
123   * @author Weston Shakespear
124   * @return String The value of the complex number as a string
125   */
126  @Override
127  public String toString() {
128      String realStr = this.real.toString();
129      String imagStr = this.imaginary.toString();
130
131      String returnStr = "";
132
133      if (imagStr.equals(anObject:"0"))
134      {
135          returnStr = String.format(format:"%s", realStr);
136      }
137      else
138      {
139          returnStr = String.format(format:"%s + %si", realStr, imagStr);
140      }
141
142      return returnStr;
143  }
```

Review: Chapter 3 Programming

3.31 Testing Output

A screenshot of a VS Code console window. The title bar reads "c:\Users\wes\github-repos\cs2420_summer2023\Chapter3\Assignment_03_31 - VS Code Console". The console output is as follows:

```
Starting tests
Starting tests
Ending tests: No red output in eclipse means success
Press any key to continue . . .
```