

1.

```
package week02.lab;

Windsurf: Refactor | Explain
public class RecursionStar {
    Run | Debug | Windsurf: Refactor | Explain | Generate Javadoc | X
    public static void main(String[] args) {
        System.out.println(starString(n: 3));
    }

    Windsurf: Refactor | Explain | Generate Javadoc | X
    public static String starString(int n) {
        if (n < 0) {
            throw new IllegalArgumentException();
        }
        if (n == 0) {
            return "*";
        }
        else {
            return starString(n - 1) + starString(n - 1);
            //2^n = 2 * 2^(n-1), so double the starString
        }
    }
}
```

2.

```
1 package week02.lab;
2
3 Windsurf: Refactor | Explain
4 public class RecursionNums {
5     Run | Debug | Windsurf: Refactor | Explain | Generate Javadoc | X
6     public static void main(String[] args) {
7         writeNums(n: 5);
8     }
9
10    Windsurf: Refactor | Explain | Generate Javadoc | X
11    public static void writeNums(int n) {
12        if (n < 1) {
13            throw new IllegalArgumentException();
14        }
15        if (n == 1) {
16            System.out.print(1);
17        }
18        else {
19            writeNums(n - 1); // recursive call first so numbers print in ascending order
20            System.out.print(", " + n); // if you want descending order, this line goes first
21        }
22    }
23 }
```

PROBLEMS 27 OUTPUT DEBUG CONSOLE TERMINAL TEST RESULTS PORTS GITLENS ⚙ Run: RecursionNums +

- duey@MacBook-Air-Duy CS211 % /usr/bin/env /Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin/Contents/Ho...ders/w3/0x9k0j1s65gczs9x9ztnngbw0000gn/T/cp\_b1gly17jggzf1hn5ifl2kq1xy.jar week02.lab.RecursionNums 1, 2, 3, 4, 5
- duey@MacBook-Air-Duy CS211 %

3.

4.

```
1 package week02.lab;
2
3 Windsurf: Refactor | Explain
4 public class RecursionEvens {
5     Run | Debug | Windsurf: Refactor | Explain | Generate Javadoc | X
6     public static void main(String[] args) {
7         int product = multiplyEvens[n: 4];
8         System.out.println(product);
9     }
10
11 Windsurf: Refactor | Explain | Generate Javadoc | X
12     public static int multiplyEvens(int n) {
13         if (n <= 0) {
14             throw new IllegalArgumentException();
15         }
16         if (n == 1) {
17             return 2;
18         }
19         else {
20             return multiplyEvens(n - 1) * (2 * n);
21         }
22     }
}

```

PROBLEMS 29 OUTPUT DEBUG CONSOLE TERMINAL TEST RESULTS PORTS GITLENS ⚙ Run: RecursionEvens

- duey@MacBook-Air-Duy CS211 % /usr/bin/env /Library/Internet Plug-Ins/JavaAppletPlugin.plugin/Contents/Holders/w3/0x9k0j1s65gczs9x9ztnngbw0000gn/T/cp\_24312wvgv5vmxwc5gf71z7ej5.jar week02.lab.RecursionEvens  
384
- duey@MacBook-Air-Duy CS211 %

5.

```
Windsurf: Refactor | Explain
3   public class RecursionReverse {
4     Run | Debug | Windsurf: Refactor | Explain | Generate Javadoc | X
5       public static void main(String[] args) {
6         System.out.println(isReverse("CSE143", "341esc"));
7       }
8
9       Windsurf: Refactor | Explain | Generate Javadoc | X
10      public static boolean isReverse(String s1, String s2) {
11        if (s1.length() != s2.length()) { // if lengths don't match
12          return false;
13        }
14        if (s1.length() <= 1) { // empty or single character
15          return true;
16        }
17        char first = Character.toLowerCase(s1.charAt(0)); // first character of s1 lowercased
18        char last = Character.toLowerCase(s2.charAt(s2.length() - 1)); // last character of s2 lowercased
19        if (first != last) { // if outer character don't match
20          return false;
21        }
22        return isReverse(s1.substring(1), s2.substring(0, s2.length() - 1));
23      }
24    }
25  }
26
27 }

PROBLEMS 29 OUTPUT DEBUG CONSOLE TERMINAL TEST RESULTS PORTS GITLENS ⚙ Run: RecursionReverse

duey@MacBook-Air-Duy CS211 % /usr/bin/env /Library/Internet\ Plug-Ins/JavaAppletPlugin.plugin/Contents/Hom
ders/w3/0x90k0j1s65gczs9x9ztnngbw0000gn/T/cp_24312wgvg5vmxwc5gf71z7ej5.jar week02.lab.RecursionReverse
true
duey@MacBook-Air-Duy CS211 %
```

6.

```
3 public class RecursiveDigits {
4     Run | Debug | Windsurf: Refactor | Explain | Generate Javadoc | X
5     public static void main(String[] args) {
6         System.out.println(evenDigits(-123456789));
7     }
8
9     Windsurf: Refactor | Explain | Generate Javadoc | X
10    public static int evenDigits(int n) {
11        boolean isNegative = n < 0;
12        n = Math.abs(n);
13
14        int result = evenDigitsHelper(n); // call evenDigitHelper (recursive method)
15
16        // if statement makes sure -0 is not returned as -0 but as 0
17        // e.g. evenDigits(-7), isNegative = true, evenDigitsHelper(7) returns 0
18        // return isNegative ? -result : result would return -0
19        if (result == 0) {
20            return 0;
21        }
22
23        return isNegative ? -result : result; // if isNegative true, return -result, else return result
24    }
25
26    Windsurf: Refactor | Explain | Generate Javadoc | X
27    private static int evenDigitsHelper(int n) {
28        if (n == 0) {
29            return 0;
30        }
31
32        int digit = n % 10; // get last digit, e.g. 1234 % 10 = 4
33        int remainingDigits = evenDigitsHelper(n / 10);
34        // get remaining digits, e.g. 1234 / 10 = 123
35        // recursively call evenDigitHelper(n / 10)
36
37        if (digit % 2 == 0) { // if digit is even
38            return remainingDigits * 10 + digit; // keeps digit & appends to right of remainingDigits
39            // remainingDigits = 123, digit = 4
40            // 123 * 10 + 4 = 1234
41            // if we did 123 + 4, would we get 127, 4 wouldn't be appended
42        }
43        else { // if digit is odd, return but don't append
44            return remainingDigits;
45        }
46    }
47 }
48 }
```

I'll make sure to do 7 and 8 later in the week when I have the time but I want to try to get this week's assignment finished asap :)