

URL to GitHub Repository: <https://github.com/ZStricklin?tab=repositories>

URL to Public Link of your Video: <https://youtu.be/a1SNizMtQ6U>

Instructions:

1. Follow the **Coding Steps** below to complete this assignment.

- In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed.
- Create a new repository on GitHub for this week's assignment and push your completed code to this dedicated repo.
- Create a video showcasing your work:
 - In this video: record and present your project verbally while showing the results of the working project.
 - Easy way to Create a video: Start a meeting in Zoom, share your screen, open Eclipse with the code and your Console window, start recording & record yourself describing and running the program showing the results.
 - Your video should be a maximum of 5 minutes.
 - Upload your video with a public link.
 - Easy way to Create a Public Video Link: Upload your video recording to YouTube with a public link.

2. In addition, please include the following in your Coding Assignment Document:

- The URL for this week's GitHub repository.
- The URL of the public link of your video.

3. Save the Coding Assignment Document as a .pdf and do the following:

- Push the .pdf to the GitHub repo for this week.
 - Upload the .pdf to the LMS in your Coding Assignment Submission.
-

Coding Steps — Arrays and Methods

- Create an array of int called ages that contains the following values: 3, 9, 23, 64, 2, 8, 28, 93.
 - Programmatically subtract the value of the first element in the array from the value in the last element of the array (i.e. do not use ages[7] in your code). Print the result to the console.
 - Add a new age to your array and repeat the step above to ensure it is dynamic (works for arrays of different lengths).
 - Use a loop to iterate through the array and calculate the average age. Print the result to the console.
- Create an array of String called names that contains the following values: "Sam", "Tommy", "Tim", "Sally", "Buck", "Bob".

- Use a loop to iterate through the array and calculate the average number of letters per name. Print the result to the console.
- Use a loop to iterate through the array again and concatenate all the names together, separated by spaces, and print the result to the console.
- How do you access the last element of any array?
- How do you access the first element of any array?
- Create a new array of int called nameLengths. Write a loop to iterate over the previously created names array and add the length of each name to the nameLengths array.
- Write a loop to iterate over the nameLengths array and calculate the sum of all the elements in the array. Print the result to the console.
- Write a method that takes a String, word, and an int, n, as arguments and returns the word concatenated to itself n number of times. (i.e. if I pass in "Hello" and 3, I expect the method to return "HelloHelloHello").
- Write a method that takes two Strings, firstName and lastName, and returns a full name (the full name should be the first and the last name as a String separated by a space).
- Write a method that takes an array of int and returns true if the sum of all the ints in the array is greater than 100.
- Write a method that takes an array of double and returns the average of all the elements in the array.
- Write a method that takes two arrays of double and returns true if the average of the elements in the first array is greater than the average of the elements in the second array.
- Write a method called willBuyDrink that takes a boolean isHotOutside, and a double moneyInPocket, and returns true if it is hot outside and if moneyInPocket is greater than 10.50.
- Create a method of your own that solves a problem. In comments, write what the method does and why you created it.

```

1 package arraysandmethods;
2
3 import java.util.Arrays;
4
5 public class CodingAssignment {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         // Create an array of int called ages that contains the following values: 3, 9, 23, 64, 2, 8, 28, 93}
10        int[] ages = {3, 9, 23, 64, 2, 8, 28, 93};
11
12        // Programmatically subtract the value of the first element in the array from the value in the
13        // last element of the array (i.e. do not use ages[7] in your code). Print the result to the
14        // console.
15
16        System.out.println(ages[ages.length - 1] - ages[0]);
17
18        // Add a new age to your array and repeat the step above to ensure it is dynamic (works for
19        // arrays of different lengths).
20
21        ages[ages.length - 1] = 98;
22        for (int i = 0; i < ages.length; i++) {
23            System.out.println(ages[i]);
24        }
25
26        int n = ages.length;
27        int newAges[] = new int[n+1];
28        int value = 98;
29        System.out.println(Arrays.toString(ages));
30        for(int i = 0; i < n; i++) {
31            newAges[i] = ages[i];
32        }
33        newAges[n] = value;
34        System.out.println(Arrays.toString(newAges)); // how to print array
35        System.out.println(newAges[newAges.length - 1] - newAges[0]);
36
37        ages = newAges;
38
39        // Use a loop to iterate through the array and calculate the average age. Print the result to
40        // the console.
41
42        double sum = 0;
43        for (double number : ages) {
44            sum += number;
45        }
46        double average = sum / ages.length;
47        System.out.println(average);
48
49        // Create an array of String called names that contains the following values: "Sam", "Tommy",
50        // "Tim", "Sally", "Buck", "Bob".
51
52        String[] names = {"Sam", "Tommy", "Tim", "Sally", "Buck", "Bob"};
53
54
55
56        // Use a loop to iterate through the array and calculate the average number of letters per name.
57        // Print the result to the console.
58
59        double sumOfNamesLength = 0;
60        for (String name : names) {
61            sumOfNamesLength += name.length;
62        }
63        System.out.println("Sum of all names length: " + sumOfNamesLength);
64
65        double averageNamesLength = sumOfNamesLength / names.length;
66
67        System.out.println("Average name length: " + averageNamesLength);
68
69
70        // Use a loop to iterate through the array again and concatenate all the names together,

```

```

58
59 double sumOfNamesLength = 0;
60 for (String name : names) {
61     sumOfNamesLength += names.length;
62 }
63 System.out.println("Sum of all names length: " + sumOfNamesLength);
64
65 double averageNamesLength = sumOfNamesLength / names.length;
66
67 System.out.println("Average name length: " + averageNamesLength);
68
69
70 // Use a loop to iterate through the array again and concatenate all the names together,
71 // separated by spaces, and print the result to the console.
72 //Name Placeholder.
73 String listOfNames = "";
74 for (String name : names) {
75     listOfNames = listOfNames + name + " ";
76 }
77
78 // loop over names array
79 // name placeholder + " "
80 // print out placeholder
81 System.out.println(listOfNames + ".");
82
83 // How do you access the last element of any array?
84
85 System.out.println(names[names.length - 1]);
86
87 // How do you access the first element of any array?
88
89 System.out.println(names[0]);
90 // Create a new array of int called nameLengths. Write a loop to iterate over the previously
91 // created names array and add the length of each name to the nameLengths array.
92
93
94 // calculate the length of names array values
95 // an int [] nameLengths == what does it equal though
96
97 int [] nameLengths = new int[names.length];
98
99 for (int i = 0; i < names.length; i++) {
100     nameLengths[i] = names[i].length();
101 }
102
103 System.out.println("Name lengths are: " + nameLengths[i]);}
104
105
106 //int[] nameLengths = {};
107 //int nameLength = 0;
108 //for (int i = 0; i < names.length; i++) {
109 //nameLengths[i]= names.length;
110 //
111 //System.out.println(nameLengths[i]);
112
113
114
115 // Write a loop to iterate over the nameLengths array and calculate the sum of all the elements
116 // in the array. Print the result to the console.
117 //
118 int sumOfNamesLengthArray = 0;
119 for (int length : nameLengths) {
120     sumOfNamesLengthArray += nameLengths.length;
121 }
122 System.out.println("Sum of names length 2nd array: " + sumOfNamesLengthArray);
123
124 // Write a method that takes a String, word, and an int, n, as arguments and returns the word
125 // concatenated to itself n number of times. (i.e. if I pass in "Hello" and 3, I expect the
126 // method to return "HelloHelloHello").
127
128 // method 1

```

```

128 // method 1
129 System.out.println(concatenanteWord("Zack", 4));
130
131
132 // Write a method that takes two Strings, firstName and lastName, and returns a full name (the
133 // full name should be the first and the last name as a String separated by a space).
134
135 //method 2 - fullName
136 // public static String fullName(String firstName, String lastName)
137 // return = firstName + " " + LastName
138 System.out.println(fullName("Zack","Stricklin"));
139
140 // Write a method that takes an array of int and returns true if the sum of all the ints in the
141 // array is greater than 100.
142 // Method 3
143 // public static int greaterThanHundred(int sum, boolean min)
144 //     int integer = 1
145 //     for (int i = 0; i <= min; i++) {
146 //         integer = int[i]
147 //     } if (integer >= 100) {
148 //         return
149 //     }
150 int[] newIntArray = {2, 4, 9, 85};
151 System.out.println(greaterThanHundred(newIntArray));
152
153 // Write a method that takes an array of double and returns the average of all the elements in
154 // the array.
155
156 //Method 4
157 //Double array average
158 double[] doubles = {1.23, 3.45, 3.12, 2.34};
159 System.out.println(doubleAverage(doubles));
160
161
162 // Write a method that takes two arrays of double and returns true if the average of the
163 // elements in the first array is greater than the average of the elements in the second array.
164 // Method 5
165
166 //public static boolean greaterThan(double [] a, double [] b) {
167 //
168 double[] greaterTest = {8.9, 5.4, 3.2, 9.8};
169 System.out.println(greaterThan(greaterTest, doubles));
170
171 // Write a method called willBuyDrink that takes a boolean isHotOutside, and a double
172 // moneyInPocket, and returns true if it is hot outside and if moneyInPocket is greater than
173 // 10.50.
174 //Method 6
175
176 boolean isHotOutside = true;
177 double moneyInPocket = 10.75;
178 System.out.println(willBuyDrink(isHotOutside,moneyInPocket));
179
180
181
182 //
183
184
185 // Create a method of your own that solves a problem. In comments, write what the method does
186 // and why you created it.
187 // Method 7
188
189 //If cook time is less than 10 print "Bit too crunchy!" if cook time is greater than
190 //10 print "Al Dente!" --
191 int cookTime = 12;
192 System.out.print(cookingSpaghetti(cookTime));
193
194 } // method
195
196 // Method 1 Concatenante word by int number of times
197 public static String concatenanteWord(String word, int n) {
197 public static String concatenanteWord(String word, int n) {
198 String result = "";

```

```

199 for (int index = 1; index <= n; index++) {
200     result = result + word;
201 }
202 return result;
203
204 }
205
206 // Method 2
207 public static String fullName(String firstName, String lastName) {
208     return firstName + " " + lastName;
209 }
210
211 // Method 3
212 public static boolean greaterThanHundred(int[] sum) {
213     int integer = 0;
214     for(int index = 0; index < sum.length; index++) {
215         integer += sum[index];
216     }
217     // for every integer called "number" (or any variable) in the array called " sum " do something
218     // integer = integer + number (same as integer += number)
219     //for (int number : sum) {
220     // integer += number ;
221     //}
222
223     //for(int i = 0; i < sum; i++) {
224     //integer = sum[i];
225     //}
226     //if (integer > 100) {
227     //return true;
228     //}
229     //else {
230     //return false;
231     //}
232     return integer > 100 ;
233 }
234
235 //Method 4 average of double
236 public static double doubleAverage(double[] array) {
237     double sum = 0;
238     for (double number : array) {
239         sum += number;
240     }
241 }
242 double average = sum / array.length;
243 return average;
244 }
245
246 // Method 5 - boolean a greater than b.
247
248 public static boolean greaterThan(double[] a, double[] b) {
249     double arrayA = 0;
250     double arrayB = 0;
251     for (double number : a) {
252         arrayA += number;
253     }
254     double averageA = arrayA / a.length;
255
256     for (double number : b) {
257         arrayB += number;
258     }
259     double averageB = arrayB / b.length;
260     return averageA > averageB;
261 }
262 // return a > b
263
264 //Method 6
265 public static boolean willBuyDrink(boolean isHotOutside, double moneyInPocket) {
266     //

```

```
267 if (isHotOutside != false && moneyInPocket > 10.50) {
268
269     return true;
270 }
271 else {
272     return false; |
273 }
274
275 }
276
277 //Method 7 - I'm terrible at cooking spaghetti.
278 public static String cookingSpaghetti(int cookTime) {
279     String cookedEnough = "Al Dente!";
280     String notCookedEnough = "Bit too crunchy!";
281
282
283     if (cookTime < 10) {
284         return notCookedEnough;
285     }
286     else if (cookTime > 10) {
287         return cookedEnough;
288     }
289     return notCookedEnough;
290 }
291
292
```