



## Week 4 Coding Assignment

**URL to GitHub Repository:**

**URL to Public Link of your Video:**

---

### 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.
- 

<https://youtu.be/1k1jW3na8FQ> Week4Coding Assignment Video

<https://github.com/mtomasi17/week4CodingAssignment.git> Github Repository



## Week 4 Coding Assignment

### Coding Steps — Arrays and Methods

1. Create an array of int called ages that contains the following values: 3, 9, 23, 64, 2, 8, 28, 93.
  - a. 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.
  - b. Add a new age to your array and repeat the step above to ensure it is dynamic (works for arrays of different lengths).
  - c. Use a loop to iterate through the array and calculate the average age. Print the result to the console.
2. Create an array of String called names that contains the following values: “Sam”, “Tommy”, “Tim”, “Sally”, “Buck”, “Bob”.
  - a. Use a loop to iterate through the array and calculate the average number of letters per name. Print the result to the console.
  - b. 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.
3. How do you access the last element of any array?
4. How do you access the first element of any array?
5. 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.
6. 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.
7. 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”).
8. 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).
9. 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.
10. Write a method that takes an array of double and returns the average of all the elements in the array.
11. 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.
12. 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.
13. Create a method of your own that solves a problem. In comments, write what the method does and why you created it.



## Week 4 Coding Assignment

```
codingAssignment.java x
1 package com.promineotech;
2
3 import java.util.Arrays;
4
5 public class codingAssignment {
6
7     public static void main(String[] args) {
8
9         //1. Create an array of int called ages that contains the following values: 3, 9, 23, 64, 2, 8, 2
10        int[] ages = {3, 9, 23, 64, 2, 8, 28, 93};
11
12        /* a. Programmatically subtract the value of the first element in the array from the value in the
13         * of the array (i.e. do not use ages[7] in your code). Print the result to the console.
14         */
15        int firstElement = ages[0];
16        int lastElement = ages[ages.length - 1];
17        int result = lastElement - firstElement;
18        System.out.println(result);
19
20        // b. Add a new age to your array and repeat the step above to ensure it is dynamic (works for arrays of different lengths).
21        ages = Arrays.copyOf(ages, ages.length + 1);
22        ages[ages.length - 1] = 80;
23        firstElement = ages[0];
24        lastElement = ages[ages.length - 1];
25        result = lastElement - firstElement;
26        System.out.println(result);
27
28        // c. Use a loop to iterate through the array and calculate the average age. Print the result to the console.
29        int sum = 0;
30        for (int i = 0; i < ages.length; i++) {
31            sum += ages[i];
32        }
33        double averageAge = (double) sum / ages.length;
34        System.out.println(averageAge);
35
36        //2. Create an array of String called names that contains the following values: "Sam", "Tommy", "Tim", "Sally", "Buck", "Bob".
37        String [] names = {"Sam", "Tommy", "Tim", "Sally", "Buck", "Bob"};
38
39        /* a. Use a loop to iterate through the array and calculate the average number of letters per name. Print the result
40         * to the console.
41         */
42        int totalLetters = 0;
43        for (String name : names) {
```

```
Problems Javadoc Declaration Console x
<terminated> codingAssignment [Java Application] C:\Program Files\Java\jdk-17\bin\javac
90
77
34.444444444444444
3.8333333333333335
Sam Tommy Tim Sally Buck Bob
23
HelloHelloHello
Michael Tomasi
true
6.6
true
true
Keep Studying! You are Almost done.
```

```
codingAssignment.java x
42        int totalLetters = 0;
43        for (String name : names) {
44            totalLetters += name.length();
45        }
46        double averageLetters = (double)totalLetters / names.length;
47
48        System.out.println(averageLetters);
49        /* b. Use a loop to iterate through the array again and concatenate all the names together, separated by spaces, and
50         * print the result to the console.
51         */
52        String concatenatedNames = "";
53        for (String name : names) {
54            concatenatedNames += name + " ";
55        }
56        System.out.println(concatinatedNames);
57
58        //3. How do you access the last element of any array?
59        /*Answer: for the above code the way we would access the last element we could do the following:
60         * String lastElement = names[names.length - 1];
61         * we access it by using the length property - 1
62         */
63
64        //4. How do you access the first element of any array?
65        /*Answer: for the above code the way we would access the first element we could do the following:
66         * String firstElement = names[0];
67         * we access it by using the index number [0]
68         */
69
70        /*5. Create a new array of int called nameLengths. Write a loop to iterate over the previously created names array and
71         * add the length of each name to the nameLengths array.
72         */
73        int [] nameLengths = new int[names.length];
74        for (int i = 0; i < names.length; i++) {
75            nameLengths[i] = names[i].length();
76        }
77
78        /*6. Write a loop to iterate over the nameLengths array and calculate the sum of all the elements in the array.
79         * Print the result to the console
80         */
81        int nameLengthsSum = 0;
82        for (int i = 0; i < nameLengths.length; i++) {
83            nameLengthsSum += nameLengths[i];
84        }
85        System.out.println(nameLengthsSum);
86
87
```



## Week 4 Coding Assignment

```
codingAssignment.java x
82 System.out.println(nameLengthsSum);
83
84 //7. Method Test
85 String word = "Hello";
86 int n = 3;
87 String concatenatedWord = concatenateWord(word, n);
88 System.out.println(concatenatedWord);
89
90 //8. Method Test
91 String firstName = "Michael";
92 String lastName = "Tomasi";
93 String fullName = genFullName(firstName, lastName);
94 System.out.println(fullName);
95
96 //9. Method Test
97 int[] numbers = {15, 26, 30, 45};
98 boolean isSumGreaterThan100 = isSumGreaterThan100(numbers);
99 System.out.println(isSumGreaterThan100);
100
101 //10. Method Test
102 double[] doubleNumbers = {2.5, 4.8, 6.9, 9.1, 9.7};
103 double average = calculateAverage1(doubleNumbers);
104 System.out.println(average);
105
106 //11. Method Test
107 double[] array1 = {2.5, 4.8, 6.3, 9.1, 1.7};
108 double[] array2 = {1.2, 3.9, 5.6, 7.8, 2.4};
109 boolean isFirstArrayAverageGreater = compareArrayAverages(array1, array2);
110 System.out.println(isFirstArrayAverageGreater);
111
112 //12. Method Test
113 boolean isHotOutside = true;
114 double moneyInPocket = 25.00;
115 boolean willBuyDrink = willBuyDrink(isHotOutside, moneyInPocket);
116 System.out.println(willBuyDrink);
117
118 //13. Method Test
119 boolean workNotCompleted = true;
120 int hoursOfStudy = 15;
121 String schoolWorkStatus = schoolWorkStatus(workNotCompleted, hoursOfStudy);
122 System.out.println(schoolWorkStatus);
123
124
125
126 }
127
128 /*13. Create a method of your own that solves a problem. In comments, write what the method does and why you created it.
129 * This Method takes a boolean value known as workNotCompleted and an int value of hours of study and returns a string
130 * based on whether your work is completed and how many hours you have worked. This would provide a user with some "Verbal"
131 * input to encourage them or give advice.
132 */
133 public static String schoolWorkStatus(boolean workNotCompleted, int hoursOfStudy) {
134     if (workNotCompleted && hoursOfStudy > 20) {
135         return "You Should Seek Help With Classwork!";
136     } else if (workNotCompleted && hoursOfStudy < 20) {
137         return "Keep Studying! You are Almost done.";
138     } else if (!workNotCompleted && hoursOfStudy > 20) {
139         return "Good Job!";
140     } else {
141         return "You are a fast learner! Good Job!";
142     }
143 }
144
145 /*12. Write a method called willBuyDrink that takes a boolean isHotOutside, and a double moneyInPocket, and returns
146 * true if it is hot outside and if moneyInPocket is greater than 10.50
147 */
148 public static boolean willBuyDrink(boolean isHotOutside, double moneyInPocket) {
149     if (isHotOutside && moneyInPocket > 10.50) {
150         return true;
151     } else {
152         return false;
153     }
154 }
155
156 }
157
158 /* 11. Write a method that takes two arrays of double and returns true if the average of the elements in the first array
159 * is greater than the average of the elements in the second array.
160 */
161 private static boolean compareArrayAverages(double[] array1, double[] array2) {
162     double average1 = calculateAverage1(array1);
163     double average2 = calculateAverage1(array2);
164     return average1 > average2;
165 }
166
167 private static double calculateAverage1(double[] array) {
168     double sum = 0;
169     for (double num : array) {
```



## Week 4 Coding Assignment

```
codingAssignment.java X
166 private static double calculateAverage1(double[] array) {
167     double sum = 0;
168     for (double num : array) {
169         sum += num;
170     }
171     return sum / array.length;
172 }
173
174 }
175 //10. Write a method that takes an array of double and returns the average of all the elements in the array.
176 public static double calculateAverage(double[] doubleNumbers) {
177     double sum = 0;
178     for (double num : doubleNumbers) {
179         sum += num;
180     }
181     return sum / doubleNumbers.length;
182 }
183
184 //9. 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.
185 public static boolean isSumGreaterThan100(int[] numbers) {
186     int sum = 0;
187     for (int num : numbers) {
188         sum += num;
189     }
190     return sum > 100;
191 }
192
193 }
194 /*8. Write a method that takes two Strings, firstName and lastName, and returns a full name (the full name should
195 * be the first and the last name as a String separated by a space).
196 */
197 public static String genFullName(String firstName, String lastName) {
198     return firstName + " " + lastName;
199 }
200
201 }
202
203
204 /*7. Write a method that takes a String, word, and an int, n, as arguments and returns the word concatenated to
205 * itself n number of times. (i.e. if I pass in "Hello" and 3, I expect the method to return "HelloHelloHello")
206 */
207 public static String concatenateWord(String word, int n) {
208     StringBuilder sb = new StringBuilder();
209
210 }
211
212 }
213
214 }
215 }
216
```

```
204 /*7. Write a method that takes a String, word, and an int, n, as arguments and returns the word concatenated to
205 * itself n number of times. (i.e. if I pass in "Hello" and 3, I expect the method to return "HelloHelloHello")
206 */
207 public static String concatenateWord(String word, int n) {
208     StringBuilder sb = new StringBuilder();
209     for (int i = 0; i < n; i++) {
210         sb.append(word);
211     }
212     return sb.toString();
213 }
214
215 }
```