

# Intro to Java Week 3 Coding Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

**Instructions:** 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. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your Java project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

## Coding Steps:

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?

By using the length of the array minus 1

4. How do you access the first element of any array?

By using the [0] on the name of the 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 would 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.

### **Screenshots of Code:**

```

1 package assignment;
2
3 public class week3Assignment {
4
5     public static void main(String[] args) {
6
7         // question 1 of creating an array of integers
8
9         int[] ages = {3, 9, 23, 64, 2, 8, 28, 93, 44}; // 44 added to the given array
10
11         int firstMinusLast = ages[ages.length - 1] - ages[0];
12
13         System.out.println(firstMinusLast); // printing 41 on the screen
14
15         double sumOfAges = 0;
16         for (int i = 0; i < ages.length; i++) {
17             sumOfAges += ages[i];
18         }
19
20         double averageOfAges = sumOfAges / ages.length;
21
22         System.out.println(averageOfAges); // printing 30.44444444444443 on the screen
23
24
25         // question 2a) printing the average characters of an array of strings
26
27         String[] names = {"Sam", "Tommy", "Tim", "Sally", "Buck", "Bob"};
28         double totalNamesCharacters = 0;
29
30         for (String name : names) {
31             totalNamesCharacters += name.length();
32         }
33
34         double averageNumbesOfNames = totalNamesCharacters / names.length;
35
36         System.out.println(averageNumbesOfNames); // printing 3.8333333333335 on the screen
37
38         // Question 2b) concatenating the values of the names array
39
40         String concatenationOfNames = "";
41
42         for (String name : names) {
43             concatenationOfNames += name + " ";
44         }
45
46         System.out.println(concatanationOfNames); // Printing "Sam Tommy Tim Sally Buck Bob"
47

```

```

4/
48 // Question 5 Creating the nameLengths array
49
50 int[] nameLengths = new int[6];
51
52 for (int i =0; i < names.length; i++) {
53     nameLengths[i] = names[i].length();
54 }
55
56 // Question 6 Sum of element in the nameLengths Array
57
58 int sumOfNameLengths = 0;
59 for (int element : nameLengths) {
60     sumOfNameLengths += element;
61 }
62
63 System.out.println(sumOfNameLengths); // printing 23 on the screen
64
65 // Question 7: testing the output of the multiplyAString method
66
67 System.out.println(multiplyAString("Hello", 3));
68
69 // Question 8: testing the output of the createFullName method
70
71 System.out.println(createFullName("John", "Smith")); // printing "John Smith"
72
73
74 // Question 9 : testing the output of the greaterThan100 method
75
76 int[] range = {12, 35, 34, 1};
77
78 System.out.println(greaterThan100(range)); // printing false
79
80 // Question 10 and 11
81
82 double[] values = {12.1, 32.0, 5, 0.9};
83
84 double[] numbers = {23.0, 35.6, 46.4};
85
86 System.out.println(averageOfAnArrayOfDouble(values)); // printing 12.5 for question 10
87
88 System.out.println(isAverageOfFirstArrayGreaterThenAverageOfSecondArray(numbers, values)); // Printing true for question 11
89
90 // Question 12: testing the output of the willBuyDrink method
91
92 System.out.println(willBuyDrink(true, 9.6) ); // printing false since 9.6 is less than 10.5
93
94
95 // Question 13: checking the existInTheArray method using the string array names and a string "Senghor"
96
97 System.out.println(existInTheArray(names, "Senghor")); // printing false
98
99 }
100
101

```

```

102 // method of question 7
103
104 public static String multiplyAString (String word, int n ) {
105     String result = "";
106     for (int i = 1; i <= n ; i++) {
107         result += word;
108     }
109     return result;
110 }
111
112 // method of question 8
113
114 public static String createFullName (String firstName, String lastName) {
115     return firstName + " " + lastName;
116 }
117
118 // method of question 9
119
120 public static boolean greaterThan100 (int[] array) {
121     int sum = 0;
122     for (int value : array) {
123         sum += value;
124     }
125     return sum > 100;
126 }
127
128

```

```

129 // method of question 10
130
131 public static double averageOfAnArrayOfDouble (double[] array) {
132     double sum = 0;
133     for (double element : array) {
134         sum += element;
135     }
136     return sum / array.length;
137 }
138
139 // Method of question 11
140
141 public static boolean isAverageOfFirstArrayGreaterThanOrEqualToSecondArray (double[] firstArray, double[] secondArray) {
142     return averageOfAnArrayOfDouble(firstArray) >= averageOfAnArrayOfDouble(secondArray);
143 }
144
145 // Method of question 12
146
147 public static boolean willBuyDrink (boolean isHotOutside, double moneyInPocket) {
148
149     if (isHotOutside && moneyInPocket > 10.5 ) {
150         return true;
151     }
152     else {
153         return false;
154     }
155 }
156

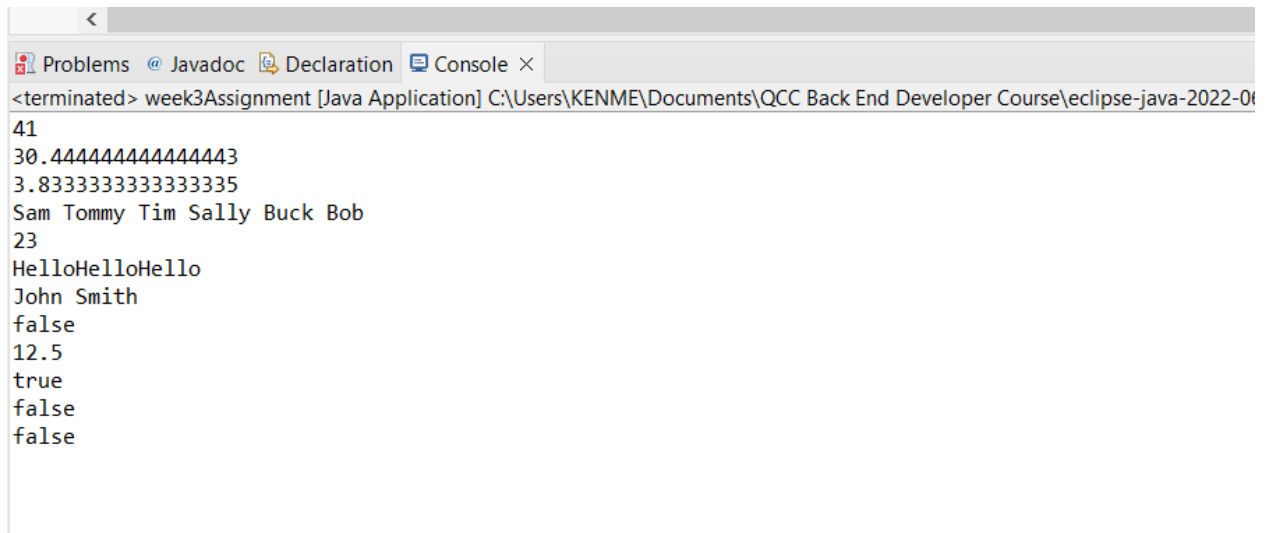
```

```

157 // This method of my mine takes and array of string and check if a specified string is listed inside the array
158
159 public static boolean existInTheArray (String[] arrayOfStrings, String string ) {
160     boolean result = false;
161     for (String member : arrayOfStrings) {
162         if (member == string) {
163             result = true;
164             break;
165         }
166     }
167 }
168
169     return result;
170 }
171
172 }
173

```

## Screenshots of Running Application:



```

<terminated> week3Assignment [Java Application] C:\Users\KENME\Documents\QCC Back End Developer Course\eclipse-java-2022-04
41
30.444444444444443
3.8333333333333335
Sam Tommy Tim Sally Buck Bob
23
HelloHelloHello
John Smith
false
12.5
true
false
false

```

## URL to GitHub Repository:

<https://github.com/kenmeugnesenghor/JavaTraining-Week3>