

# 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?
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 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
2 public class Week3Homework {
3
4     public static void main(String[] args) {
5
6         //Create an array of int called ages that contains the following values: 3, 9, 23, 64, 2, 8, 28, 93
7         int[] ages = {3, 9, 23, 64, 2, 8, 28, 93, 14};
8
9         //Programmatically subtract the value of the first element in the array from the value in the last element of the array
10        //(i.e. do not use ages[7] in your code)
11        int result = ages[ages.length-1] - ages[0];
12
13        //Print the result to the console.
14        System.out.println(result); //This printed 90, as 93 - 3 is 90.
15
16        //Add a new age to your array and repeat the step above to ensure it is dynamic (works for arrays of different lengths).
17        //I went back to the array and added 14. The above code line now prints out 11, as 14 - 3 is 11.
18
19        //Use a loop to iterate through the array and calculate the average age. Print the result to the console.
20        double averageAge = 0;
21        for(int age : ages) {
22            averageAge += age;
23        }
24        averageAge = averageAge / ages.length;
25        System.out.println("Average age of the ages[] is: " + averageAge);
26
27        //Create an array of String called names that contains the following values: "Sam", "Tommy", "Tim", "Sally", "Buck", "Bob".
28        String[] names = {"Sam", "Tommy", "Tim", "Sally", "Buck", "Bob"};
29
30        //Use a loop to iterate through the array and calculate the average number of letters per name. Print the result to the console.
31        double averageNameLength = 0;
32        for(String name : names) {
33            averageNameLength += name.length();
34        }
35        averageNameLength = averageNameLength / names.length;
36        System.out.println("The average length of the names in the names[] is: " + averageNameLength);
37
38        //Use a loop to iterate through the array again and concatenate all the names together, separated by spaces,
39        //and print the result to the console.
40        String combinedWords = "";
41        for(int i = 0; i < names.length; i++) {
42            combinedWords += names[i] + " ";
43        }
44        System.out.println(combinedWords);
45
46        //How do you access the last element of any array?
47        //arrayName[arrayName.length-1]; will always refer to the last element in an array.
48
49        //How do you access the first element of any array?
50        //arrayName[0]; will always refer to the first element in an array, because arrays start at 0.
51

```

```

51
52 //Create a new array of int called nameLengths.
53 //Write a loop to iterate over the previously created names array and add the length of each name to the nameLengths array.
54 int nameLengths[] = new int[6];
55 for(int i = 0; i < nameLengths.length; i++) {
56     nameLengths[i] = names[i].length();
57     // System.out.println(nameLengths[i]); //this was just to check my work
58 }
59
60
61 //Write a loop to iterate over the nameLengths array and calculate the sum of all the elements in the array.
62 //Print the result to the console.
63 int sumOfNameLengths = 0;
64 for(int name : nameLengths) {
65     sumOfNameLengths += name;
66 }
67 System.out.println("The sum of the name lengths is: " + sumOfNameLengths);
68
69 //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.
70 //(i.e. if I pass in "Hello" and 3, I would expect the method to return "HelloHelloHello").
71 System.out.println(wordTimes("Hello", 3));
72
73 //Write a method that takes two Strings, firstName and lastName, and returns a full name
74 //(the full name should be the first and the last name as a String separated by a space).
75 System.out.println(fullName("Andrew", "Farmer"));
76
77 //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.
78 int[] bigNumbers = {50, 85};
79 System.out.println(oneHundred(bigNumbers));
80
81 //Write a method that takes an array of double and returns the average of all the elements in the array.
82 double[] averageDouble = {150, 160, 179};
83 System.out.println(averageThis(averageDouble));
84
85 //Write a method that takes two arrays of double and returns true if the average of the elements in the first array is
86 //greater than the average of the elements in the second array.
87 double[] averageDouble2 = {137, 160, 179};
88 System.out.println(isFirstBigger(averageDouble, averageDouble2));
89
90 //Write a method called willBuyDrink that takes a boolean isHotOutside,
91 //and a double moneyInPocket, and returns true if it is hot outside and if moneyInPocket is greater than 10.50.
92 System.out.println("Will you buy a drink? " + willBuyDrink(true, 20.0));
93
94 //Create a method of your own that solves a problem. In comments, write what the method does and
95 //why you created it
96 System.out.println(LetterAverage(names));
97 //I wrote this for niche scenario as to when that would be relevant.
98 //Maybe it's good for Scrabble or something, I don't know.
99
00 }
01

```

```

101
102 //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.
103 //(i.e. if I pass in "Hello" and 3, I would expect the method to return "HelloHelloHello").
104● public static String wordTimes (String word, int n) {
105     String holder = "";
106     for(int i = 0; i < n; i++)
107         holder += word;
108     return holder;
109 }
110
111 //Write a method that takes two Strings, firstName and lastName, and returns a full name
112 //(the full name should be the first and the last name as a String separated by a space).
113● public static String fullName(String firstName, String lastName) {
114     return firstName + " " + lastName;
115 }
116
117 //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.
118● public static boolean oneHundred(int[] array) {
119     int total = 0;
120     for(int i = 0; i < array.length; i++) {
121         total += array[i];
122         if(total > 100)
123             return true;
124     }
125     return false;
126 }
127
128 //Write a method that takes an array of double and returns the average of all the elements in the array.
129● public static double averageThis(double[] array) {
130     double total = 0;
131     for(double number : array) {
132         total += number;
133     }
134     return total / array.length;
135 }
136
137 //Write a method that takes two arrays of double and returns true if the average of the elements in the first array is
138 //greater than the average of the elements in the second array.
139● public static boolean isFirstBigger(double[] a, double[] b) {
140     if(averageThis(a) > averageThis(b))
141         return true;
142     return false;
143 }
144
145 //Write a method called willBuyDrink that takes a boolean isHotOutside,
146 //and a double moneyInPocket, and returns true if it is hot outside and if moneyInPocket is greater than 10.50.
147● public static boolean willBuyDrink(boolean isHotOutside, double moneyInPocket) {
148
149     if(isHotOutside == true && moneyInPocket > 10.50)
150         return true;
151     return false;
152 }

```

```

153
154 //Create a method of your own that solves a problem. In comments, write what the method does and
155 //why you created it
156● public static double letterAverage(String[] array) {
157     double x = 0;
158     for(String word : array) {
159         x+= word.length();
160     }
161     return x / array.length; //This code takes an array of Strings and returns the average number of letters per word.
162     //I wrote this for niche scenario as to when that would be relevant.
163     //Maybe it's good for Scrabble or something, I don't know.
164 }
165
166 }
167

```

### Screenshots of Running Application:

```
11
Average age of the ages[] is: 27.11111111111111
The average length of the names in the names[] is: 3.8333333333333335
Sam Tommy Tim Sally Buck Bob
The sum of the name lengths is: 23
HelloHelloHello
Andrew Farmer
true
163.0
true
Will you buy a drink? true
3.8333333333333335
```

### URL to GitHub Repository:

<https://github.com/AFarmer94/Week3Homework.git>