

# 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?

```
array[array.length - 1]);
```

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

```
array[0];
```

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 helloWorld;
2 import java.util.*;
3 import java.io.*;
4 import java.lang.reflect.Array;
5 import java.text.*;
6 import java.math.*;
7 import java.util.regex.*;
8
9 public class Week3_Coding_Assignment {
10
11     public static void main(String[] args) {
12         // 1. Create an array of int called ages that contains the following values: 3, 9, 23, 64, 2, 8, 28, 93.
13         // a. Programmatically subtract the value of the first element in the array from the value in the last element of the array
14         // (i.e. do not use ages[7] in your code). Print the result to the console.
15         // b. Add a new age to your array and repeat the step above to ensure it is dynamic (works for arrays of different lengths).
16         // c. Use a loop to iterate through the array and calculate the average age. Print the result to the console.
17
18         int[] ages = { 3, 9, 23, 64, 2, 8, 28, 93 };
19         int lengthOfArray = ages.length;
20         int difference = Math.abs(ages[lengthOfArray - 1] - ages[0]);
21         System.out.println(difference);
22
23         // New array
24         int[] newAges = new int[ages.length];
25
26         // Copy of original array to new array
27         System.arraycopy(ages, 0, newAges, 0, ages.length);
28
29         // Subtracting first value from the last
30         lengthOfArray = newAges.length;
31         newAges[lengthOfArray - 1] = 40;
32         difference = Math.abs(newAges[lengthOfArray - 1] - newAges[0]);
33         System.out.println(difference);
34
35         // Math to find average
36         int sum = 0;
37         for (int i = 0; i < newAges.length; i++) {
38             sum += newAges[i];
39         }
40         double average = sum / lengthOfArray;
41         System.out.println(average);
42
43         // 2. Create an array of String called names that contains the following values: "Sam", "Tommy", "Tim", "Sally", "Buck", "Bob".
44         // a. Use a loop to iterate through the array and calculate the average number of letters per name. Print the result to the console.
45         // 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.
46         // 3. How do you access the last element of any array?
47         // array[array.length - 1];
48         // 4. How do you access the first element of any array?
49         // array[0];
50
51         String[] names = { "Sam", "Tommy", "Tim", "Sally", "Buck", "Bob" };
52         sum = 0;
53         for (int i = 0; i < names.length; i++) {
54             sum += names[i].length();
55         }
56         int average2 = sum / names.length;
57         System.out.println(average2);
58
59         String allNames = "";
60         for (int i = 0; i < names.length; i++) {
61             allNames += names[i] + " ";
62         }
63
64         // 5. Create a new array of int called nameLengths.
65         // Write a loop to iterate over the previously created names
66
67
68
69
70
71
72
73
74

```

```

75 //      array and add the length of each name to the nameLengths array.
76 //      //////////////////////////////////
77
78      int[] nameLengths = new int[names.length];
79      for (int i = 0; i < names.length; i++) {
80          nameLengths[i] = names[i].length();}
81
82      //System.out.println(nameLengths[i]);}
83      // checking my for loop
84
85      //////////////////////////////////
86 //      6. Write a loop to iterate over the nameLengths array
87 //      and calculate the sum of all the elements in the array.
88 //      Print the result to the console.
89 //      //////////////////////////////////
90
91      sum = 0;
92      for (int i = 0; i < nameLengths.length; i++) {
93          sum += nameLengths[i] + names[i].length();}
94      System.out.println(sum);
95
96
97 //7. Print test for Method
98 System.out.println(multiplyString("Hello" , 3));
99
100
101 //8. Print test for Method
102 System.out.println(createFullName("Bob" , "Ross"));
103
104
105 //9.Print test for Method
106 System.out.println(isGreaterThan100(ages));
107
108
109 //10. Double array to test method
110 double[] grades = new double[5];
111 grades[0] = 44.3;
112 grades[1] = 75.6;
113 grades[2] = 96.4;
114 grades[3] = 76.9;
115 grades[4] = 98.1;
116
117 //10. Print test for method
118 System.out.println(findAverage(grades));
119
120
121 //11. Duplicated Double Array with grades[1] changed for test method
122 double[] grades1 = new double[5];
123 grades[0] = 44.3;
124 grades[1] = 0;
125 grades[2] = 96.4;
126 grades[3] = 76.9;
127 grades[4] = 98.1;
128
129 //11. Print test for method
130 System.out.println(isFirstArrayGreater(grades , grades1));
131
132 //12. Boolean and Double values to test method
133 boolean isHotOutside = true;
134 double moneyInPocket = 10;
135
136 //12. Print test for method
137 System.out.println(willBuyDrink(isHotOutside , moneyInPocket));
138
139 //13. Print test for Method
140 System.out.println(smallestNumber(ages));
141
142
143
144

```

```

142
143
144
145 }          //METHODS//
146
147
148
149
150
151 //          //
152 //          7. Write a method that takes a String, word,
153 //          and an int, n, as arguments and returns the word
154 //          concatenated to itself n number of times.
155 //          (i.e. if I pass in "Hello" and 3, I would expect
156 //          the method to return "HelloHelloHello").
157 //          //
158 public static String multiplyString(String word, int n) {
159     String result = "";
160     for (int i = 0; i < n; i++) {
161         result += word;
162     }
163     return result;
164
165     //
166 //          8. Write a method that takes two Strings,
167 //          firstName and lastName, and returns a full name (the full name
168 //          should be the first and the last name as a String separated by a space).
169 //          //
170 public static String createFullName(String x, String y) {
171     String fullName = x + " " + y;
172     return fullName;
173
174     //
175 //          9. Write a method that takes an array
176 //          of int and returns true if the sum of all
177 //          the ints in the array is greater than 100.
178 //          //
179 public static boolean isGreaterThan100(int[] array) {
180     int sum = 0;
181     for (int i = 0; i < array.length; i++) {
182         sum += array[i];
183     }
184     if (sum > 100) {
185         return true;
186     }
187     return false;
188
189     //
190 //          10. Write a method that takes
191 //          an array of double and returns
192 //          the average of all the elements in the array.
193 //          //
194 public static double findAverage(double[] array) {
195     double sum = 0;
196     for (double number : array) {
197         sum += number;
198     }
199     return sum / array.length;
200
201     //
202 //          11. Write a method that takes two arrays of double
203 //          and returns true if the average of the elements in
204 //          the first array is greater than the average of the elements
205 //          in the second array.
206 //          //
207 public static boolean isFirstArrayGreater(double[] x, double[] y) {
208     return findAverage(x) > findAverage(y);
209
210
211
212
213
214
215

```

```

213
214 public static boolean isFirstArrayGreater(double[] x, double[] y) {
215     return findAverage(x) > findAverage(y);}
216
217
218
219
220 //          //
221 //          12.      Write a method called willBuyDrink
222 //                  that takes a boolean isHotOutside, and a double moneyInPocket,
223 //                  and returns true if it is hot outside and
224 //                  if moneyInPocket is greater than 10.50.
225 //          //
226
227 public static boolean willBuyDrink(boolean isHotOutside, double moneyInPocket) {
228     if (isHotOutside == true && moneyInPocket > 10.5) {
229         return true;
230     } else {
231         return false;}}
232
233
234 //          //
235 //          13.      Create a method of your own that solves a problem.
236 //                  In comments, write what the method does and why you created it.
237
238 // This method will find the smallest number in a given array of int
239 // I created this so if you have a large data set you can identify outliers fast as this code can be flipped
240 // to be the largest number.
241
242
243 public static int smallestNumber(int[] numbers) {
244     int smallest = numbers[0];
245     for (int number : numbers) {
246         if (number < smallest) {
247             smallest = number;}}
248     return smallest;}}

```

## Screenshots of Running Application:



```

<terminated> Week3_Coding_Assignment [Java Application] C:\Program Files\Java\jdk-11.0.14\bin\javaw.exe (May 18, 2022, 8:49:13 PM - 8:49:13 PM) [pid: 2732]
90
37
22.0
3
46
HelloHelloHello
Bob Ross
true
78.26000000000002
true
false
2

```

## URL to GitHub Repository:

<https://github.com/coltonrood/Week3-coding-assignment1>