

# 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 //CodingAssignment.java X // MethodsAndArrays.java // Power.java
2 import java.lang.reflect.Array;
3
4 public class CodingAssignment {
5
6     public static void main(String[] args) {
7
8         // Coding Assignment Steps 1a, 1b, and 1c
9         int[] ages = {3, 9, 22, 64, 2, 8, 28, 93, 96};
10         int lastNumber = ages.length-1;
11         System.out.println(ages[lastNumber] - ages[0]);
12         double sum = 0;
13         for (int age : ages) {
14             sum += age;
15         }
16         System.out.println(sum / ages.length);
17
18         //Coding Assignment Steps 2a and 2b
19         String[] names = {"Sam", "Tommy", "Tim", "Sally", "Buck", "Bob"};
20         double namesLength = 0.0;
21         for (String name : names) {
22             namesLength += name.length();
23         }
24         double averageLetters = namesLength / names.length;
25         System.out.println(averageLetters);
26         for (int i = 0; i < names.length - 1; i++) {
27             System.out.print(names[i] + " ");
28         }
29
30         //Coding Assignment Step 3: array.length-1 or array[lastPosition]
31         //Coding Assignment Step 4: array[0]
32         //Coding Assignment Step 5:
33         System.out.println("");
34         int isCodingDone = 0;
35         for (String name : names) {
36             if (isCodingDone == 0) {
37                 int[] namesLengths = new int[names.length];
38                 for (int i = 0; i < names.length; i++) {
39                     namesLengths[i] = names[i].length();
40                 }
41                 for (int namesLength : namesLengths) {
42                     System.out.print(namesLength + " ");
43                 }
44             }
45             //Coding Assignment Step 6:
46             System.out.println("");
47             int allLetters = 0;
48             for (int length : namesLengths) {
49                 allLetters += length;
50             }
51             System.out.println(allLetters);
52
53             //Coding Assignment Step 7:
54             System.out.println(concatenateWords("Hello", 3));
55         }
56     }
57 }
```

```
51
52 //Coding Assignment Step 8:
53 String firstName = "Bill";
54 String lastName = "Jones";
55 System.out.println(createFullName(firstName, lastName));
56
57 //Coding Assignment Step 9:
58 double[] array = {25.0, 7.1, 4.2, 10.3};
59 System.out.println(compareSumOfNumbers(array));
60
61 //Coding Assignment Step 10:
62 double[] singleArray = {14.2, 57.9, 24.5};
63 System.out.println(findAverage(singleArray));
64
65 //Coding Assignment Step 11:
66 double[] first = {18.5, 48.5, 16.75, 9.25};
67 double[] second = {19.0, 30.0, 34.25, 22.50};
68 System.out.println(findAverageAvg(second, first));
69
70 //Coding Assignment Step 12:
71 boolean isHotOutside = true;
72 double moneyInPocket = 9.75;
73 System.out.println(shouldDrink(isHotOutside, moneyInPocket));
74
75 //Coding Assignment Step 13:
76 boolean isCodingDone = false;
77 double tempOutside = 75.0;
78 System.out.println(useFreeTime(isCodingDone, tempOutside));
79
80 //Used to make quick decisions about how to use my free time, instead of scrolling on my phone.
81
82 //Method Step 13:
83 public static String useFreeTime (boolean isCodingDone, double tempOutside) {
84     if (isCodingDone == false) {
85         return "Go your coding homework!";
86     } else if (isCodingDone == true && tempOutside < 65.0) {
87         return "Head a book!";
88     } else {
89         return "Go for a walk!";
90     }
91 }
92
93 //k tempOutside == 65.0
94 //Method Step 12:
95 public static boolean willHydrate (boolean isHotOutside, double moneyInPocket) {
96     if (isHotOutside == true && moneyInPocket > 10.50) {
97         return true;
98     } else {
99         return false;
100     }
101 }
```

```
99
100 }
101 //Method Step 11:
102 public static boolean findGreaterAvg (double[] a, double[] b) {
103     double avgA = findAverage(a);
104     double avgB = findAverage(b);
105     if (avgA > avgB) {
106         return true;
107     } else {
108         return false;
109     }
110 }
111
112 //Method Step 10:
113
114 public static double findAverage (double[] doubles) {
115     double total = 0;
116     for (double number : doubles) {
117         total += number;
118     }
119     return total / doubles.length;
120 }
121
122 //Method Step 9:
123 public static boolean compareSumOfNumbers (double[] numbers) {
124     int sum = 0;
125     for (double number : numbers) {
126         sum += number;
127     }
128     if (sum > 100) {
129         return true;
130     } else {
131         return false;
132     }
133 }
134
135 //Method Step 8:
136 public static String createFullName (String firstName, String lastName) {
137     return firstName + " " + lastName;
138 }
139
140 //Method Step 7:
141 public static String concatenateWords(String word, int n) {
142     int i = 1;
143     String output = "";
144     while (i <= n) {
145         output += word;
146     }
147     return output;
148 }
149
150 }
```

## Screenshots of Running Application:

```
Problems @ Javadoc Declaration Console X
<terminated> CodingAssignment [Java Application] /Users/caseykobosh/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.
47
31.111111111111111
3.8333333333333333
Sam Tommy Tim Sally Buck Bob
3 5 3 5 4 3
23
HelloHelloHello
Bill Jones
false
32.199999999999996
true
false
Do your coding homework!
```

## URL to GitHub Repository:

<https://github.com/CKobosh/Week-3-Coding-Assignment.git>