

# 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?

`arr[arr.length - 1]`

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

`arr[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
2 public class Week_3_Coding_Assignment {
3
4     public static void main(String[] args) {
5
6         int[] arr1 = {101, 43, 44, 12};
7         double[] arr2 = {34.5, 2, 54, 654.33};
8         double[] arr3 = {23.5, 55.5, 43, 101};
9
10        ages();
11        System.out.println("-----");
12        names();
13        System.out.println("-----");
14        System.out.println(concatWord("我叫李乐-", 5));
15        System.out.println("-----");
16        System.out.println(fullName("Lee", "Shawver"));
17        System.out.println("-----");
18        System.out.println(greaterThan100(arr1));
19        System.out.println("-----");
20        System.out.println(average(arr2));
21        System.out.println("-----");
22        System.out.println(greaterAverage(arr2, arr3));
23        System.out.println("-----");
24        System.out.println(willBuyDrink(true, 10.7));
25        System.out.println("-----");
26        System.out.println(containsString("我喜欢编程"));
27        System.out.println("-----");
28    }
29
30    public static void ages() {
31        int[] ages = {3, 9, 23, 64, 2, 8, 28, 93};
32        int firstFromLast = ages[ages.length - 1] - ages[0];
33        System.out.println(firstFromLast);
34        ages[7] = 37;
35        firstFromLast = ages[ages.length - 1] - ages[0];
36        System.out.println(firstFromLast);
37        double totalAges = 0;
38        for (int age : ages) {
39            totalAges += age;
40        }
41        double averageAge = totalAges / ages.length;
42        System.out.println(averageAge);
43    }

```

```

45 public static void names() {
46     String[] names = new String[6];
47     names[0] = "Sam";
48     names[1] = "Tommy";
49     names[2] = "Tim";
50     names[3] = "Sally";
51     names[4] = "Buck";
52     names[5] = "Bob";
53
54     int totalLetters = 0;
55     for (String name : names) {
56         totalLetters += name.length();
57     }
58     double averageLength = (totalLetters / names.length);
59     System.out.println(averageLength);
60
61     StringBuilder allNames = new StringBuilder("");
62     for (String name : names) {
63         allNames.append(name + " ");
64     }
65     System.out.println(allNames);
66     System.out.println("-----");
67
68     int[] nameLengths = new int[names.length];
69     for (int i = 0; i < names.length; i++) {
70         nameLengths[i] = names[i].length();
71     }
72     int sumOfLengths = 0;
73     for (int length : nameLengths) {
74         sumOfLengths += length;
75     }
76     System.out.println(sumOfLengths);
77 }
78
79 public static String concatWord(String word, int n) {
80     String wholeWord = "";
81     for (int i = 1; i <= n; i++) {
82         wholeWord = wholeWord + word;
83     }
84     return wholeWord;
85 }
86
87 public static String fullName(String firstName, String lastName) {
88     String fullName = firstName + " " + lastName;
89     return fullName;
90 }
91
92 public static Boolean greaterThan100(int[] numbers) {
93     int sum = 0;
94     for (int number : numbers) {
95         sum += number;
96     }
97     if (sum > 100) {
98         return true;
99     } else return false;
100 }
101
102 public static double average(double[] numbers) {
103     double total = 0;
104     for (double number : numbers) {
105         total += number;
106     }
107     double average = total / numbers.length;
108     return average;
109 }

```

```

111 public static Boolean greaterAverage(double[] arr1, double[] arr2) {
112     double average1 = average(arr1);
113     double average2 = average(arr2);
114     if (average1 > average2) {
115         return true;
116     } else return false;
117 }
118
119 public static Boolean willBuyDrink(Boolean isHotOutside, double moneyInPocket) {
120     if (isHotOutside && moneyInPocket > 10.5) {
121         return true;
122     } else return false;
123 }
124
125 public static Boolean containsString(String words) {
126     // this method checks to see if a given string contains the characters "编程"
127     // which is Chinese for programming
128     if (words.contains("编程")) {
129         return true;
130     } else return false;
131 }
132 }

```

### Screenshots of Running Application:

```

90
34
21.75
-----
3.0
Sam Tommy Tim Sally Buck Bob
-----
23
-----
我叫李乐-我叫李乐-我叫李乐-我叫李乐-我叫李乐-
-----
Lee Shawver
-----
true
-----
186.2075
-----
true
-----
true
-----
true
-----

```

### URL to GitHub Repository: