

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:

```
121 // Coding Step 8: Write a method that takes 2 Strings firstName and lastName, and returns a full name
122 // (the full name should be the first and the last name as a String separated by a space).
123 System.out.println("\nCoding Step 8: Calls concatFirstNameLastName method.");
124
125     concatFirstNameLastName("Tom","Jerry");
126
127
128 // Coding Step 9: Write a method that takes an array of int and returns true if the sum of all the int in the array is greater than 100.
129 System.out.println("\nCoding Step 9: Calls intArray.");
130
131     int[] intArray = {10, 5, 33, 17, 1, 48};
132
133     intArrayGreaterThan100(intArray);
134
135 // Coding Step 10: Write a method that takes an array of double and returns the average of all the elements in the array.
136 System.out.println("\nCoding Step 10: Calls doubleNumbersArray.");
137
138     double[] doubleNumbersArray = {22.4, 77, 15.9, 2.3, 62.1, 33.0, 19.76};
139
140     avgOfDoubleArray(doubleNumbersArray);
141
142
143 // Coding Step 11: Write a method that takes two arrays of double and returns true if the average of the elements in the first array
144 // is greater than the average of the elements in the second array.
145 System.out.println("\nCoding Step 11: Calls method avgOfFirstArrayGreaterThanAvgOfSecondArray.");
146
147     double[] firstArrays = {17.5, 13.76, 9.43, 2.2};
148     double[] secondArrays = {22.3, 5.4, 2, 3.3, 6.997};
149
150     avgOfFirstArrayGreaterThanAvgOfSecondArray(firstArrays, secondArrays);
151
152
153 // Coding Step 12: Write a method called willBuyDrink that takes a boolean isHotOutside and a double moneyInPocket and returns true
154 // if it is hot outside and if moneyInPocket is greater than 10.50.
155 System.out.println("\nCoding Step 12: Calls method willBuyDrink");
156
157     boolean isHotOutside = true;
158     double moneyInPocket = 11.50;
159
160     willBuyDrink(isHotOutside, moneyInPocket);
161
162
163 // Coding Step 13: Write a method of your own that solves a problem.
164 // Create a method called operatingMarginOutcome that takes double called totalRevenue and double called totalExpense
165 // and calculates operatingMarginPercent. If operatingMarginPercent is greater than 5% prints out to console "Good Job!" and operating margin percent.
166 // Otherwise it prints to console "Room for improvement." and lists operating margin percent. Note, only include 2 digits after decimal for percentage.
167 // I wrote this because financial calculations are a huge part of any business. Operating Margin being one of those. It tells a story of how the company is doing.
168
169 System.out.println("\nCoding Step 13: Calls method operatingMarginOutcome");
170 double totalRevenue = 400000;
171 double totalExpense = 375000;
172 double netIncome = (totalRevenue - totalExpense);
173
174 System.out.println(operatingMarginOutcome(netIncome, totalRevenue));
175
176
177
178
179 }
```

```
180 /** METHODS BELOW *****/
181
182 //Coding Step 7: Method repeats Alive concatenated to itself 5 times.
183 public static String repeatWord(String word, int n) {
184     String concatWord = "";
185     for(int i = 1; i<=n; i++) {
186         concatWord += word;
187     }
188     return concatWord;
189 }
190
191
192 //Coding Step 8: Method that concatenates 2 Strings firstName and lastName, separated with a space.
193 public static String concatFirstNameLastName(String firstName, String lastName) {
194     String fullName = firstName + " " + lastName;
195     return fullName;
196 }
197
198
199 //Coding Step 9: Method that returns true if sum of array "intArray" is greater than 100.
200 public static boolean intArrayGreaterThan100(int[] intArray) {
201     int counter = 0;
202     boolean result = true;
203
204     for(int sumArray : intArray) {
205         counter += sumArray;
206         if(counter>100) {
207             result = true;
208         } else {
209             result = false;
210         }
211     }
212     return result;
213 }
214
215
216 //Coding Step 10: Method sums double array and returns the avg of elements of array.
217 public static double avgOfDoubleArray(double[] doubleNumbersArray) {
218
219     double sumOfDoubleNumbersArray = 0;
220
221     for(double doubleNumber : doubleNumbersArray) {
222         sumOfDoubleNumbersArray += doubleNumber;
223     }
224     return sumOfDoubleNumbersArray / doubleNumbersArray.length;
225 }
226
227
228 //Coding Step 11: Method takes two arrays of double and returns true if avg of elements in 1st array > avg of elements in 2nd array.
229 public static boolean avgOfFirstArrayGreaterThanAvgOfSecondArray(double[] firstArrays, double[] secondArrays) {
230     double sumOfFirstArray = 0;
231     double sumOfSecondArray = 0;
232     //boolean result;
233
234     for(double firstArray : firstArrays) {
235         sumOfFirstArray += firstArray;
236     }
237     for(double secondArray : secondArrays) {
238         sumOfSecondArray += secondArray;
239     }
240 }
```

```

240
241         return (sumOfFirstArray / firstArrays.length) > (sumOfSecondArray / secondArrays.length);
242     }
243
244
245 //Coding Step 12: Write a method called willBuyDrink that takes a boolean isHotOutside, and a double moneyInPocket,
246 // and returns true if it is hot outside and if moneyInPocket is greater than 10.50.
247 public static boolean willBuyDrink(boolean isHotOutside, double moneyInPocket) {
248
249     if(isHotOutside && moneyInPocket > 10.50) {
250         return true;
251     } else {
252         return false;
253     }
254 }
255
256
257 //Coding Step 13: Write a method of your own that solves a problem.
258 // Create a method called operatingMarginOutcome that takes double called totalRevenue and double called totalExpense
259 // and calculates operatingMarginPercent. If operatingMarginPercent is greater than 5% prints out to console "Good Job!" and displays operating margin percent.
260 // Otherwise it prints to console "Room for improvement." and displays operating margin percent.
261 // I wrote this because financial calculations are a huge part of any business. Operating Margin being one of those. It tells a story of how the company is doing.
262
263 public static String operatingMarginOutcome(double netIncome, double totalRevenue) {
264
265     if(netIncome / totalRevenue > .05) {
266         //return ("Good Job! Operating Margin " + ((netIncome / totalRevenue)*100) + "%");
267         return String.format("Good Job! Operating Margin is %.2f%%", (netIncome / totalRevenue)*100);
268     }else {
269         return String.format("Room for improvement. Operating Margin is %.2f%%", (netIncome / totalRevenue)*100);
270     }
271 }
272
273 /** STOP *****//
274 }

```

<<<<End. See Next Page>>>>

Screenshots of Running Application : I did not print if it didn't say to.

```
<terminated> Assignment [Java Application] C:\Program Files\Java\jdk-11.0.13\bin\javaw.exe (Mar 12, 2022, 7:28:52 PM - 7:28:52 PM)
Coding Step 1: **Create an array of int called ages that contains values: 3, 9, 23, 64, 2, 8, 28, 93.**
Coding Step 1a: **Subtract the value of the first element from the value of the last element.**
90
Coding Step 1b: **Add a new age to your array and repeat the step above. **
73
Coding Step 1c: **Using a loop calculate the average age (Note: I used this age array {3,9,23,64,2,8,28,93}).**
28.75
Coding Step 2a: **Use a loop to iterate through the array and calculate the average number of letters per name. Print to the console.**
3.8333333333333335
Coding Step 2b: **Use a loop to iterate through the array again and concatenate all the names together, separated by spaces, and print to the console.**
Sam Tommy Tim Sally Buck Bob
Coding Step 3: **How do you access the last element of any array? Note: I created a new String array called farmAnimalsArray{"Cat", "Dog", "Horse", "Cow", "Pig"}.**
Pig
Coding Step 4: **How do you access the last element of any array? Note: I created a new String array called farmAnimalsArray{"Cat", "Dog", "Horse", "Cow", "Pig"}.**
Cat
Coding Step 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.**
Coding Step 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.**
29
Coding Step 7: Calls repeatWord method.
Coding Step 8: Calls concatFirstNameLastName method.
Coding Step 9: Calls intArray.
Coding Step 10: Calls doubleNumbersArray.

Coding Step 11: Calls method avgOffFirstArrayGreaterThanAvgOfSecondArray.
Coding Step 12: Calls method willBuyDrink
Coding Step 13: Calls method operatingMarginOutcome
Good Job! Operating Margin is 6.25%
```

URL to GitHub Repository:

[Java-Week3-CodingAssignment/Assignment.java at main · LynnaeInama/Java-Week3-CodingAssignment \(github.com\)](https://github.com/LynnaeInama/Java-Week3-CodingAssignment/blob/main/Assignment.java)

Screenshots of Running Application – showing everything printing to console:

This allows you to see every question. Though I don't believe this is what you are asking for, but I provided just in case.

```
<terminated> Assignment [Java Application] C:\Program Files\Java\jdk-11.0.13\bin\javaw.exe (Mar 12, 2022, 6:19:16 PM – 6:19:16 PM)
Coding Step 1: **Create an array of int called ages that contains values: 3, 9, 23, 64, 2, 8, 28, 93.**
3
9
23
64
2
8
28
93

Coding Step 1a: **Subtract the value of the first element from the value of the last element.**
90

Coding Step 1b: ***Add a new age to your array and repeat the step above. ***
73

Coding Step 1c: ***Using a loop calculate the average age (Note: I used this age array {3,9,23,64,2,8,28,93}).***
28.75

Coding Step 2a: ***Use a loop to iterate through the array and calculate the average number of letters per name. Print to the console.***
3.8333333333333335

Coding Step 2b: ***Use a loop to iterate through the array again and concatenate all the names together, separated by spaces, and print to the console.***
Sam Tommy Tim Sally Buck Bob

Coding Step 3: ***How do you access the last element of any array? Note: I created a new String array called farmAnimalsArray{"Cat", "Dog", "Horse", "Cow", "Pig"}.***
Pig

Coding Step 4: ***How do you access the last element of any array? Note: I created a new String array called farmAnimalsArray{"Cat", "Dog", "Horse", "Cow", "Pig"}.***
Cat

Coding Step 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.***
4
6
4
6
5
4

Coding Step 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.***
29

Coding Step 7: Calls repeatWord method.
AliveAliveAliveAliveAlive

Coding Step 8: Calls concatFirstNameLastName method.
Tom Jerry

Coding Step 9: Calls intArray.
true

Coding Step 10: Calls doubleNumbersArray.
33.20857142857143

Coding Step 11: Calls method avgOfFirstArrayGreaterThanAvgOfSecondArray.
true

Coding Step 12: Calls method willBuyDrink
true

Coding Step 13: Calls method operatingMarginOutcome
Good Job! Operating Margin is 6.25%
```