

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 (do not type ANY numbers in the operation, `ages[7] - ages[0]` is not allowed). 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:

```
*learnJavaWeekThree.java
1 package learnJavaWeekThree;
2 public class learnJavaWeekThree {
3
4     public static void main(String[] args) {
5
6         //=====
7         // 1.) Create an array of int called ages that contains the following values: 3, 9, 23, 64, 2, 8, 28, 93.
8         int[] ages = { 3, 9, 23, 64, 2, 8, 28, 93 };
9
10        // a.) Subtract the value of the first element in the array from the value in the last element of the array.
11        int firstElement = ages[0];
12        int lastElement = ages[ages.length - 1];
13        int ageSubtracted = lastElement - firstElement;
14        System.out.println("1a.) The first element in the array, subtracted by the last element is: "+ ageSubtracted);
15
16        // b.) Add a new age to your array and repeat the step above to ensure it is dynamic.
17        int[] moreAges = { 3, 9, 23, 64, 2, 8, 28, 93, 45};
18
19        firstElement = moreAges[0];
20        lastElement = moreAges[moreAges.length - 1];
21        ageSubtracted = lastElement - firstElement;
22        System.out.println("1b.) The first element in the second longer array, subtracted by the last element is: "+ ageSubtracted + " . . . neat!");
23
24        // c.) Use a loop to iterate through the array and calculate the average age.
25        int sum = 0;
26        int count = 0;
27        for ( int age : ages ) {
28            sum += age;
29            count ++;
30        }
31        int average = sum / count;
32        System.out.println("1c.) The average of the array is around ~" +average + "\n");
33
34        // =====
35        // 2.) Create an array of String called names that contains the following values: "Sam", "Tommy", "Tim", "Sally", "Buck", "Bob".
36        String[] names = { "Sam", "Tommy", "Tim", "Sally", "Buck", "Bob" };
37
38        // a.) Use a loop to iterate through the array and calculate the average number of letters per name. Print the result to the console.
39        int totalCharacters = 0;
40        int nameCount = 0;
41        for ( String name : names ) {
42            // Create an char array of given String
43            char[] ch = name.toCharArray();
44            int wordLength = ch.length;
45            float averageChars = 0.0f;
46
47            // average number of characters = total characters / quantity of names
48            totalCharacters += wordLength;
49            nameCount++;
50            averageChars = totalCharacters / nameCount;
51            System.out.println("2a.) After processing " + name + " the average number of characters per name is now about ~" + averageChars);
52        }
53    }
```

```
54
55 // 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
56 String largeName = "";
57 for (String name : names) {
58     largeName += (name + " ");
59 }
60 System.out.println("2b.) " + largeName);
61
62
63 //=====
64 // 3.) How do you access the last element of any array?
65 System.out.println("\n3.) One accesses the last element in an array by calling the element within the array that's \n found at the .length - 1 position");
66
67 // 4.) How do you access the first element of any array?
68 System.out.println("4.) Because arrays begin at 0, one accesses the first element in an array by calling the element within the array that's \n found at index 0");
69
70 // 5.) Create a new array of int called nameLengths. Write a loop to iterate over the previously created names array
71 // and add the length of each name to the nameLengths array.
72
73
74 int[] nameLengths = new int[names.length];
75 //String strArray[] = new String[nameLengths.length];
76 int arrayPosition = 0;
77 System.out.println("\n5.) The following name lengths have been added to the 'nameLengths' array:");
78 for (String name : names) {
79     char[] ch = name.toCharArray();
80     int wordLength = ch.length;
81     nameLengths[arrayPosition] = wordLength;
82     System.out.println(name[arrayPosition] + " - name length: " + nameLengths[arrayPosition]);
83     arrayPosition += 1;
84 }
85
86 // 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.
87 int nameLengthSum = 0;
88 for (int nameLength : nameLengths) {
89     nameLengthSum += nameLength;
90 }
91 System.out.println("\n6.) The sum of all nameLengths would be " + nameLengthSum + ".");
92
93 // 7.) Write a method that takes a String, word, and an int, n, as arguments and returns the word concatenated to
94 // itself n number of times.
95 System.out.println("\n7.) " + stringMultiplier("word", 3));
96
97 // 8.) Write a method that takes two Strings, firstName and lastName, and returns a full name.
98 System.out.println("\n8.) " + concatName("Tylor", "Davis"));
99
100 // 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.
101 System.out.println("\n9.) The answer for this particular int array is: " + sumChecker(nameLengths));
102
103 // 10.) Write a method that takes an array of double and returns the average of all the elements in the array.
104 double[] testArray = {34.5, 98.60, 9.33, 9.90, 42.00};
105 System.out.println("\n10.) The average of all elements within this array of double is: " + averageDoubleArray(testArray));
106
```

```
// 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.
double[] testArrayOne = {73.55, 67.60, 9.33, 876.98, 42.00} ;
double[] testArrayTwo = {34.50, 87.83, 0.00, 9.90, 2.00} ;
System.out.println("\n11.) Is the average of the elements in the first double array greater than the second?: " + compareArrays(testArrayOne, testArrayTwo));

// 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.
System.out.println("\n12.) It's a good idea to get milk?: " + willBuyDrink( true, 11.20));

// 13.) Create a method of your own that solves a problem. In comments, write what the method does and why you created it.
System.out.println("\n13.) the input string backwards is: " + stringReverser( "I love Teresa"));
}
// Custom method, for number 13. Inverses the order of the characters within the string.
static String stringReverser( String inputString) {
    // Create an char array of given String
    char[] ch = inputString.toCharArray();
    // Initialize outputString;
    String outputString = "";
    // Go through the characters within the given input string, from last to first, and concatenate it to the output String
    for (int i = ch.length -1; (i >= 0) ; i--) {
        outputString += ch[i];
    }
    return outputString;
}

// Will Buy Milk? from number twelve.
static boolean willBuyDrink( boolean isHotOutside , double moneyInPocket) {

    if (isHotOutside && moneyInPocket >= 10.50) {
        return true;
    }
    else {
        return false;
    }
}

// Compares the averages of arrays, for number eleven.
static boolean compareArrays(double[] testArray1, double[] testArray2 ) {
    double arrayOneAverage = averageDoubleArray(testArray1);
    double arrayTwoAverage = averageDoubleArray(testArray2);

    if (arrayOneAverage > arrayTwoAverage)
        return true;
    else {
        return false;
    }
}

// Double array Average.... go for number ten.
static double averageDoubleArray(double[] inputArray) {
    double doubleSum = 00.00;
    for (double element : inputArray) {
```



```
static boolean compareArrays(double[] testArray1, double[] testArray2 ) {
    double arrayOneAverage = averageDoubleArray(testArray1);
    double arrayTwoAverage = averageDoubleArray(testArray2);

    if (arrayOneAverage > arrayTwoAverage)
        return true;
    else {
        return false;
    }
}

// Double array Average.... eg for number ten.
static double averageDoubleArray(double[] inputArray) {
    double doubleSum = 00.00;
    for (double element : inputArray) {
        doubleSum += element;
    }
    double average = doubleSum / inputArray.length;
    return average;
}

// Checks the sum of int arrays
static boolean sumChecker( int[] arrayOfNumbers) {
    int intArraySum = 0;
    for (int i : arrayOfNumbers) {
        intArraySum += arrayOfNumbers[i];
    }
    if (intArraySum >= 100) {
        return true;
    }
    else {
        return false;
    }
}

// String multiplier for number seven.
static String stringMultiplier( String word , int multiplier) {
    String largeWord = "";
    for (int i = 1; i <= multiplier; i++ ) {
        largeWord += (word + " ");
    }
    return largeWord;
}

//String builder for number eight.
static String concatName( String firstName , String lastName) {
    return firstName + " " + lastName;
}
```

Screenshots of Running Application:

```
Problems @ Javadoc Declaration Console X
<terminated> learnJavaWeekThree [Java Application] C:\Program Files\Java\jre1.8.0_161\bin\javaw.exe (Feb 21, 2020, 2:30:47 PM)
1a.) The first element in the array, subtracted by the last element is: 90
1b.) The first element in the second longer array, subtracted by the last element is: 42 . . . neat!
1c.) The average of the array is around ~28

2a.) After processing Sam the average number of characters per name is now about ~3.0
2a.) After processing Tommy the average number of characters per name is now about ~4.0
2a.) After processing Tim the average number of characters per name is now about ~3.0
2a.) After processing Sally the average number of characters per name is now about ~4.0
2a.) After processing Buck the average number of characters per name is now about ~4.0
2a.) After processing Bob the average number of characters per name is now about ~3.0
2b.) Sam Tommy Tim Sally Buck Bob

3.) One accesses the last element in an array by calling the element within the array that's
found at the .length - 1 position,
for example: arrayName[arrayName.length - 1] would access the final element in arrayName.

4.) Because arrays begin at 0, one accesses the first element in an array by calling the element within the array that's
found at the 0 position ,
for example: arrayName[0] would access the final element in arrayName.

5.) The following name lengths have been added to the 'namesLength' array:
Sam - name length: 3
Tommy - name length: 5
Tim - name length: 3
Sally - name length: 5
Buck - name length: 4
Bob - name length: 3

6.) The sum of all nameLengths would be 23.

7.) word word word

8.) Tylor Davis

9.) The answer for this particular int array is: false

10.) The average of all elements within this array of double is: 38.866

11.) Is the average of the elements in the first double array greater than the second?: true

12.) It's a good idea to get milk?: true

13.) the input string backwards is: asereT evol I
```

URL to GitHub Repository:

<https://github.com/TylorTheAnvil/learnJavaWeekThree.git>