

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:

```
Week3Lab.java x
1
2 public class Week3Lab {
3
4     public static int[] addAnotherSpace(int[] arr) {
5         //This method takes an array and returns an array with the same contents and one more space
6         int[] newArr = new int[arr.length + 1];
7         //Creates another array one space larger
8         for(int i = 0; i < arr.length; i++) {
9             newArr[i] = arr[i];
10        }
11        //Loop copies each value from initial array to new array
12        return newArr;
13        //Returns new array with one extra empty space, can replace initial array
14    }
15
16    public static String wordMultiplier(String word, int n) {
17        String newWord = "";
18        for(int i = 0; i < n; i++) {
19            newWord += word;
20        }
21        return newWord;
22    }
23    //Concatenates word n times without a space
24
25    public static String fullName(String firstName, String lastName) {
26        return firstName + " " + lastName;
27    }
28    //Concatenates firstName and lastName with a space
29
30    public static boolean moreThanHundred(int[] intArray) {
31        int sumOfArray = 0;
32        for(int numbers : intArray) {
33            sumOfArray += numbers;
34        }
35        return (sumOfArray > 100);
36    }
37    //Returns true if the sum of elements in a given int array is more than 100
38
39    public static double doubleAverage(double[] doubleArray) {
40        double arraySum = 0;
41        for(double numbers : doubleArray) {
42            arraySum += numbers;
43        }
44        return arraySum / doubleArray.length;
45    }
46    //Finds average of elements in a double array
47 }
```

```

47
48● public static boolean arrayComparison(double[] array1, double[] array2) {
49    double average1 = doubleAverage(array1);
50    double average2 = doubleAverage(array2);
51    return (average1 > average2);
52}
53//Utilizing previous method, returns true if the average of first array is more than that of second array
54
55● public static boolean willBuyDrink(boolean isHotOutside, double moneyInPocket) {
56    return (isHotOutside && moneyInPocket > 10.5);
57}
58//returns true if isHotOutside is true and moneyInPocket is more than 10.5
59
60● public static String[] stringAdd(String[] initArray, String extraWord) {
61    //This method takes an already initialized string array full of values and adds another word to a newly created space
62    String[] newArray = new String[initArray.length + 1];
63    for (int i = 0; i < initArray.length; i++) {
64        newArray[i] = initArray[i];
65    }
66    newArray[newArray.length - 1] = extraWord;
67    return newArray;
68    //By setting the string used as initArray equal to this, it adds a word onto a string, which would not be
69    //possible previously if all spaces were taken up already
70}
71
72
73● public static void main (String[] args) {
74
75    //Code block below follows requirements of Step #1
76    int[] ages = {3, 9, 23, 64, 2, 8, 28, 93};
77    System.out.println("Last minus first: " + (ages[ages.length - 1] - ages[1]));
78    ages = addAnotherSpace(ages);
79    ages[8] = 42;
80    //Accomplishing instructions by using new method
81    System.out.println("Last minus first: " + (ages[ages.length - 1] - ages[1]));
82    double ageTotal = 0;
83    for(int age : ages) {
84        ageTotal += age;
85    }
86    System.out.println("Average age: " + ageTotal / ages.length);
87    //Enhanced for loop practice above
88    //ageTotal is a double bc if it was an int, it would create a rounding error line 30
89

```

```

89
90    //Code block below follows requirements of Step #2
91    String[] names = new String[6];
92    names[0] = "Sam";
93    names[1] = "Tommy";
94    names[2] = "Tim";
95    names[3] = "Sally";
96    names[4] = "Buck";
97    names[5] = "Bill";
98    //Using the same initialization as the int array doesn't seem to work for strings
99    //Creating the array this way takes up more space and will ideally be improved later
100    double letterCount = 0;
101    for(String name : names) {
102        letterCount += name.length();
103    }
104    System.out.println("Average name length: " + letterCount / names.length);
105    //Although not necessary in this case, letterCount is a double to avoid rounding
106    String nameConc = "";
107    for(String name : names) {
108        nameConc += name;
109        nameConc += " ";
110    }
111    System.out.println("Concatenation of all names: " + nameConc);
112    //Could be done in the first loop, but kept separate for grading clarity
113
114    //Answer to 3, for array named "arrayName" it is arrayName[arrayName.length - 1];
115    //Answer to 4, for array named "arrayName" it is arrayName[0];
116
117    //Code block below follows requirements of Step #5
118    int[] nameLengths = new int[6];
119    for(int i = 0; i < nameLengths.length; i++) {
120        nameLengths[i] = names[i].length();
121    }
122

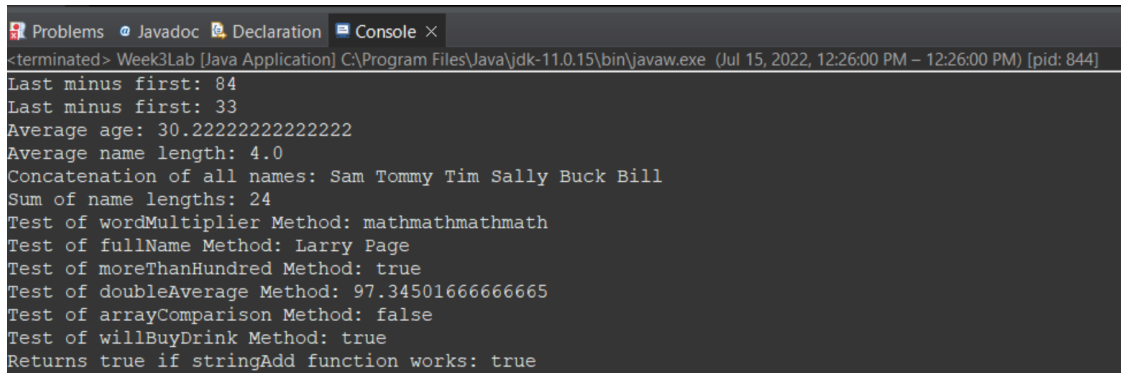
```

```

123 //Code block below follows requirements of Step #6
124 int sumOfLengths = 0;
125 int j = 0;
126 while (j < nameLengths.length) {
127     sumOfLengths += nameLengths[j];
128     j++;
129 }
130 System.out.println("Sum of name lengths: " + sumOfLengths);
131 //Using a variety of loops to practice each kind
132
133 //Code block below follows requirements of Step #7
134 System.out.println("Test of wordMultiplier Method: " + wordMultiplier("math", 4));
135
136 //Code block below follows requirements of Step #8
137 System.out.println("Test of fullName Method: " + fullName("Larry", "Page"));
138
139 //Code block below follows requirements of Step #9
140 int[] intTest = {6, 23, 4, 56, 11, 0, 1};
141 System.out.println("Test of moreThanHundred Method: " + moreThanHundred(intTest));
142
143 //Code block below follows requirements of Step #10
144 double[] doubleTest = {6.2, 400.23, 64.64, 100.0, 2.0001, 11.0};
145 System.out.println("Test of doubleAverage Method: " + doubleAverage(doubleTest));
146
147 //Code block below follows requirements of Step #11
148 double[] compareArray = {11.2, 2000.1, 2.2, 3.3, 1.1, 4.4, 5.5, 6.6, 7.7, 8.8};
149 System.out.println("Test of arrayComparison Method: " + arrayComparison(doubleTest, compareArray));
150
151 //Code block below follows requirements of Step #12
152 System.out.println("Test of willBuyDrink Method: " + willBuyDrink(true, 12.1));
153
154 //Code block below follows requirements of Step #13
155 String[] wordString = new String[3];
156 wordString[0] = "one";
157 wordString[1] = "two";
158 wordString[2] = "three";
159 wordString = stringAdd(wordString, "four");
160 System.out.println("Returns true if stringAdd function works: " + (wordString[wordString.length - 1] == "four"));
161 //stringAdd here takes an already full string array, adds another space, and assigns the new word to that space
162 //The output confirms this by comparing the final item in this array to the word that was supposed to be added
163
164
165 }
166
167 }

```

Screenshots of Running Application:



```

Problems Javadoc Declaration Console X
<terminated> Week3Lab [Java Application] C:\Program Files\Java\jdk-11.0.15\bin\javaw.exe (Jul 15, 2022, 12:26:00 PM – 12:26:00 PM) [pid: 844]
Last minus first: 84
Last minus first: 33
Average age: 30.22222222222222
Average name length: 4.0
Concatenation of all names: Sam Tommy Tim Sally Buck Bill
Sum of name lengths: 24
Test of wordMultiplier Method: mathmathmathmath
Test of fullName Method: Larry Page
Test of moreThanHundred Method: true
Test of doubleAverage Method: 97.34501666666665
Test of arrayComparison Method: false
Test of willBuyDrink Method: true
Returns true if stringAdd function works: true

```

URL to GitHub Repository:

<https://github.com/kopatsis/PromineoWeek3Assignment>