fIntro 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?

   By using the length of the array minus 1

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

   By using the [0] on the name of the 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:

```java
package assignment;

public class week3Assignment {

    public static void main(String[] args) {

        // question 1 of creating an array of integers

        int[] ages = {3, 9, 23, 64, 2, 8, 28, 93, 44}; // 44 added to the given array

        int firstMinusLast = ages[ages.length -1] - ages[0];

        System.out.println(firstMinusLast); // printing 41 on the screen

        double sumOfAges = 0;
        for (int i = 0; i < ages.length; i++) {
            sumOfAges += ages[i];
        }

        double averageOfAges = sumOfAges / ages.length;

        System.out.println(averageOfAges); // printing 30.4444444444443 on the screen



        // question 2a) printing the average characters of an array of strings

        String[] names = {"Sam", "Tommy", "Tim", "Sally", "Buck", "Bob"};
        double totalNamesCharacters = 0;

        for (String name : names) {
            totalNamesCharacters += name.length();
        }

        double averageNumbesOfNames = totalNamesCharacters / names.length;

        System.out.println(averageNumbesOfNames); // printing 3.833333333335 on the screen


        // Question 2b) concatenating the values of the names array

        String concatanationOfNames = "";

        for (String name : names) {
            concatanationOfNames += name + " ";
        }

        System.out.println(concatanationOfNames); // Printing "Sam Tommy Tim Sally Buck Bob"
```

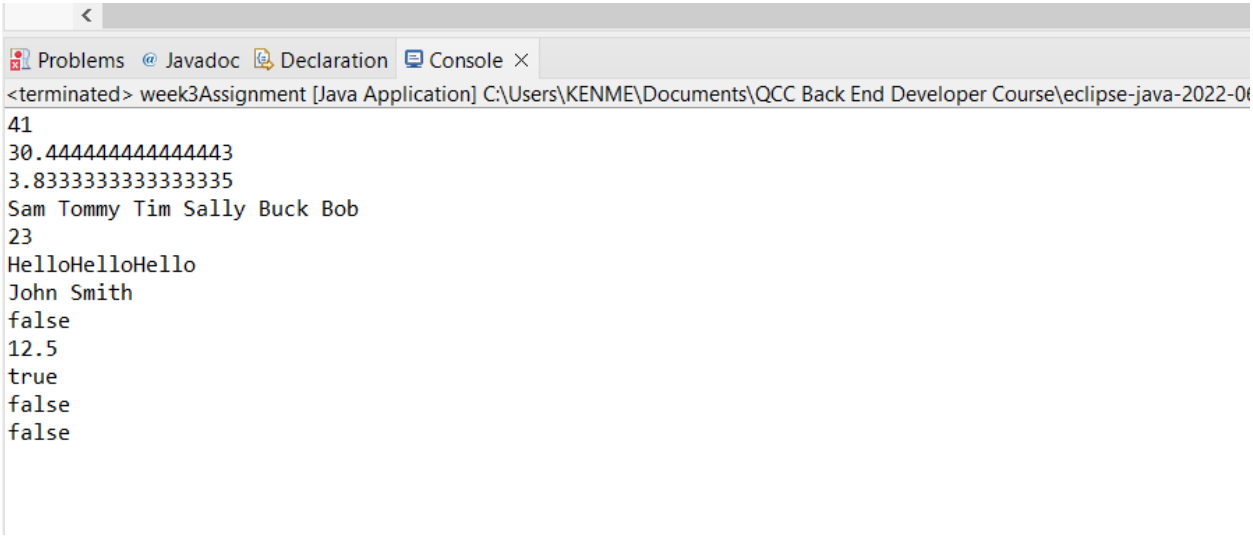
```
47
48          // Question 5 Creating the nameLengths array
49
50          int[] nameLengths = new int[6];
51
52          for (int i =0; i < names.length; i++) {
53              nameLengths[i] = names[i].length();
54          }
55
56          // Question 6 Sum of element in the nameLengths Array
57
58          int sumOfNameLengths = 0;
59          for (int element : nameLengths) {
60              sumOfNameLengths += element;
61          }
62
63          System.out.println(sumOfNameLengths); // printing 23 on the screen
64
65          // Question 7: testing the output of the multiplyAString method
66
67          System.out.println(multiplyAString("Hello", 3));
68
69          // Question 8: testing the output of the createFullName method
70
71          System.out.println(createFullName("John", "Smith")); // printing "John Smith"
72
74          // Question 9 : testing the output of the greaterThan100 method
75
76          int[] range = {12, 35, 34, 1};
77
78          System.out.println(greaterThan100(range)); // printing false
79
80          // Question 10 and 11
81
82          double[] values = {12.1, 32.0, 5, 0.9};
83
84          double[] numbers = {23.0, 35.6, 46.4};
85
86          System.out.println(averageOfAnArrayOfDouble(values)); // printing 12.5 for question 10
87
88          System.out.println(isAverageOfFirstArrayGreaterThanAverageOfSecondArray(numbers, values)); // Printing true for question 11
89
90          // Question 12: testing the output of the willBuyDrink method
91
92          System.out.println(willBuyDrink(true, 9.6) ); // printing false since 9.6 is less than 10.5
93
94
95          // Question 13: checking the existInTheArray method using the string array names and a string "Senghor"
96
97          System.out.println(existInTheArray(names, "Senghor")); // printing false
98
99
100     }
101
```

```java
    // method of question 7

    public static String multiplyAString (String word, int n ) {
           String result = "";
           for (int i = 1; i <= n ; i++) {
               result += word;
           }
           return result;

       }

    // method of question 8

    public static String createFullName (String firstName, String lastName) {
        return firstName + " " + lastName;
    }

    // method of question 9

    public static boolean greaterThan100 (int[] array) {
        int sum = 0;
        for (int value : array) {
            sum += value;
        }
        return sum > 100;
    }
```

```java
    // method of question 10

    public static double averageOfAnArrayOfDouble (double[] array) {
        double sum = 0;
        for (double element : array) {
            sum += element;
        }
        return sum / array.length;
    }

    // Method of question 11

    public static boolean isAverageOfFirstArrayGreaterThanAverageOfSecondArray (double[] firstArray, double[] secondArray) {
        return averageOfAnArrayOfDouble(firstArray) > averageOfAnArrayOfDouble(secondArray);
    }

    // Method of question 12

    public static boolean willBuyDrink (boolean isHotOutside, double moneyInPocket) {

        if (isHotOutside && moneyInPocket > 10.5 ) {
            return true;
        }
        else {
            return false;
        }
    }
```

```java
    // This method of my mine takes and array of string and check if a specified string is listed inside the array

    public static boolean existInTheArray (String[] arrayOfStrings, String string ) {
        boolean result = false;
        for (String member : arrayOfStrings) {
            if (member == string) {
                result = true;
                break;
            }
                
        }

        return result;
    }

}

```

Screenshots of Running Application:



URL to GitHub Repository:

https://github.com/kenmeugnesenghor/JavaBackEndTraining-Week3.git