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:

```java
package week3Arrays;

public class week3CodingAssignment {

    public static void main(String[] args) {
        //Question 1
        //Create an array of int called ages that contains the following values: 3, 9, 23, 64, 2, 8, 28, 93
        //print the difference of the item in the last element of the array (length of ages - 1)
        //from the item in the first element

        int[] ages = {3, 9, 23, 64, 2, 8, 28, 93};
        int difference = ages[ages.length-1] - ages[0];
        System.out.println(difference);
        //loop to print iterate through the array and calculate the average age
        double sum = 0.0;
        for (int i = 0; i < ages.length; i++) {
            sum += ages[i];
        }
        double average = sum / ages.length;
        System.out.println(average);

        //Question 2
        //Create an array of String called names that contains "Sam, Tommy, Tim, Sally, Buck and Bob."
        //Use a loop to iterate through the array and calculate the average number of letters per name.

        String[] names = {"Sam", "Tommy", "Tim", "Sally", "Buck", "Bob"};
        double numLetters = 0.0; //total number of letters for the Strings in the array
        for (int i = 0; i < names.length; i++) {
            //loop through to add the lengths of each string in the array to numLetters
            numLetters += names[i].length();
        }
        double stringAverage = numLetters / names.length;
        System.out.println(stringAverage);

        //Loop to iterate through the array again and concatenate all the names together, separated by spaces,

        String concatenate = ""; //new String to take up all the strings in one
        for (int i = 0; i < names.length; i++) {
            //loop through to add the actual strings in the array to the string cocatenate
            concatenate = concatenate + names[i] + " ";
        }
        System.out.println(concatenate);

        //Question 3
        //To access the last element of any array, do arrayName[arrayname.length - 1];

        //Question 4
        //To access the first element of any array, do arrayName[0];

```

```java
        //Question 5
        //Create a new array of int called nameLengths. Loop over the previously created names
        //array and add the length of each name to the nameLengths array.

        int[] nameLengths = new int [names.length];
        for (int i = 0; i < names.length; i++) {
            nameLengths[i] = names[i].length();
        }

        //Question 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.

        int sumNameLengths = 0;
        for (int i =0; i < nameLengths.length; i++) {
            sumNameLengths += nameLengths[i];
        }
        System.out.println(sumNameLengths);
    }

    //Question 7
    //Write a method that takes String word and int n as arguments and will loop to add
    //each word in the iteration of the loop to String complete
    //will return the String complete

    static String completeWord(String word, int n) {
        String complete = "";
        for (int i = 0; i < n; i++) {
            complete += word;
        }
        return complete;
    }

    //Question 8
    //Method that takes Strings firstName and lastName then returns the full name

    static String fullName(String firstName, String lastName) {
        String full = firstName + " " + lastName;
        return full;
    }

```

```java
    //Question 9
    //Method that takes an array of int and returns true if the sum of all ints
    //in the array is greater than 100

    static boolean overHundred(int[] array) {
        int sum = 0;
        for (int i = 0; i < array.length; i++) {
            sum += array[i];
        }
        if (sum > 100) {
            return true;
        } else {
            return false;
        }
    }

    //Question 10
    //Method that takes an array of double and returns the averages of all elements in the array

    static double averageDouble(double[] array) {
        double sum = 0.0;
        for (int i = 0; i < array.length; i++) {
            sum += array[i];
        }
        return sum / array.length;
    }

    //Question 11
    //Method that akes two arrays of double and returns true if average of elements in the first
    //array is greater than the average of elements in the second array

    static boolean greater(double[] firstArray, double[] secondArray) {
        double sumFirstArray = 0.0;
        double sumSecondArray = 0.0;
        for (int i = 0; i < firstArray.length; i++) {
            sumFirstArray += firstArray[i];
        }
        for (int j = 0; j < secondArray.length; j++) {
            sumSecondArray += secondArray[j];
        }
        double averageFirst = sumFirstArray / firstArray.length;
        double averageSecond = sumSecondArray / secondArray.length;
        if (averageFirst > averageSecond) {
            return true;
        } else {
            return false;
        }
    }
```

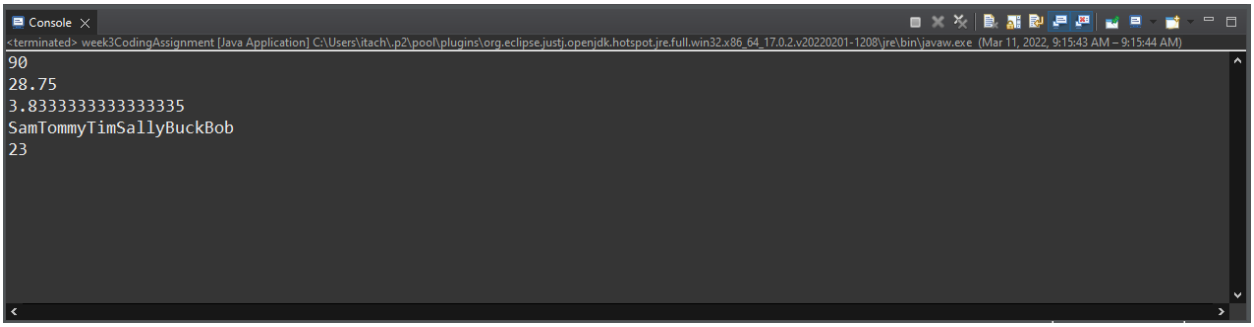
```java
    //Question 12
    //Method called willBuyDrink that takes a boolean of isHotOutside and
    //a double moneyInPocket and returns true if it is hot outside and
    //if moneyInPocket is greater than 10.50

    static boolean willBuyDrink(boolean isHotOutside, double moneyInPocket) {
        if (isHotOutside == true && moneyInPocket > 10.50) {
            return true;
        } else {
            return false;
        }
    }

    //Question 13
    //Own method to solve problem. The inflation rate in the US is soaring. This
    //method is used to return true if your new wages for the upcoming year
    //are beating the inflation rates. If yes, you should probably look for a different job.
    //This takes in 3 parameters, this year's current wage, next year's wage, and the inflation rate as a percent (eg 0.08).
    //Within the method, the inflation rate is multiplied by the current wage. If it is less than
    //next year's wage, then the method will return true. Else it will return false.

    static boolean beatInflation(double currentWage, double newWage, double inflation) {
        if (newWage > currentWage * (1 + inflation)) {
            return true;
        } else {
            return false;
        }
    }
}
```

Screenshots of Running Application:



URL to GitHub Repository:

https://github.com/AkemiTCGyt/PromineoTechWeek3CodingAssignment