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:

Main:

```java
package challenges;

public class AllChallengesAsMethods {

    public static void main(String[] args) {
        // These are the methods for the coding Challenge of Week 3

        // Test variable for sumOfIntArray
        int[] testIntAry = { 21, 13, 18, 15, 5 };

        // Test variables for avgOfDoubleArray and doubleTroubleArrays
        double[] testDblAry = { 22.50, 19.56, 32.00 };
        double[] testDblAry2 = { 1.0, 2.5, 3.6, 8.5, 10.1, 30.2 };

        // Test Variables for willBuyDrink
        double testMoneyInPocket = 10.50;
        boolean testIsHotOutside = false;

        // Test variables for properNightsRest
        int[] goToBed = { 20, 23, 0, 4, 22, 21, 2 };
        int[] wakeUp = { 7, 14, 12, 9, 6, 10, 12 };

        System.out.println("Program Start!!");
        // Method calls
        funWithIntArrays();
        funWithStringArrays();
        printingOut3and4();
        countingNameArray();
        stringIntConcat("Hello", 3);
        namingStrings("John", "Smith");
        sumOfIntArray(testIntAry);
        avgOfDoubleArray(testDblAry);
        doubleTroubleArrays(testDblAry, testDblAry2);
        willBuyDrink(testIsHotOutside, testMoneyInPocket);
        properNightsRest(goToBed, wakeUp);

        System.out.println("That's the End of the challenge!");
    }
```

Challenge 1:

```java
/**
 * funWithIntArrays
 * Creates two integer arrays Subtracts the first element from
 * the last element Calculates the average age
 */
private static void funWithIntArrays() {

    // 1
    int[] ages = { 3, 9, 23, 64, 2, 8, 28, 93 };
    // 1b
    int[] ages2 = { 3, 9, 23, 64, 2, 8, 28, 93, 32 };

    // 1a
    int element = 0;
    do {
        int answerA = ages[ages.length - 1] - ages[element];
        System.out.println("1a Answer: " + answerA);
        int answerB = ages2[ages2.length - 1] - ages2[element];
        System.out.println("1b Answer: " + answerB);
        element++;
    } while (element == 0);

    // 1c
    int sum = 0;
    for (int i = 0; i < ages.length; i++) {
        int age = ages[i];
        sum = sum + age;
    }
    int avg = sum / ages.length;
    System.out.println("1c Answer: " + avg);
}
```

Challenge 2:

```java
/**
 * funWithStringArrays
 * Creates an array of names Calculates the average number
 * of letters across all names Concatenates all of the names and prints them out
 * to the console separated by a space.
 */
private static void funWithStringArrays() {

    // 2
    String[] names = { "Sam", "Tommy", "Tim", "Sally", "Buck", "Bob" };

    // 2a
    int[] avg = new int[names.length];
    for (int i = 0; i < names.length; i++) {
        int countLetters = names[i].length();
        avg[i] = countLetters;
    }

    int sum = 0;
    for (int i = 0; i < avg.length; i++) {
        int age = avg[i];
        sum = sum + age;
    }
    int avgLtrCount = sum / avg.length;
    System.out.println("2a Answer: " + avgLtrCount);

    // 2b
    String combinedNames = "";
    for (int i = 0; i < names.length; i++) {
        String name = names[i];
        combinedNames = combinedNames + name + " ";
    }
    System.out.println("2b Answer: " + combinedNames);

}
```

Challenges 3 and 4:

```java
 /**
  * printingOut3and4
  * Prints out the answers to the questions to the console.
  */
 private static void printingOut3and4() {

     System.out.println("3 Answer: The last element of any array is accessed using (array.length - 1).");
     System.out.println("4 Answer: The first element of any array is accessed using array[0].");

 }
```

Challenges 5 and 6:

```java
/**
 * countingNameArray
 * Adds the length of each String in a predetermined String
 * array as elements in a new array Then Calculates the sum of all the elements
 * in the new array Finally prints all findings to the Console for confirmation
 */
private static void countingNameArray() {

    // String array of names from 2
    String[] names = { "Sam", "Tommy", "Tim", "Sally", "Buck", "Bob" };

    // 5. Adds the length of each String in a predetermined String array as elements
    // in a new array and prints the new array to the Console for Confirmation
    String lengthsAry = "";
    int[] nameLengths = new int[names.length];
    for (int i = 0; i < nameLengths.length; i++) {
        int countLetters = names[i].length();
        nameLengths[i] = countLetters;
        lengthsAry = lengthsAry + nameLengths[i] + " ";
    }
    System.out.println("5 Answer: " + lengthsAry);

    // 6. Iterates over the nameLengths array and calculates the sum of all the
    // elements in the array. Then Prints the total to the console
    int sum = 0;
    for (int i = 0; i < nameLengths.length; i++) {
        int age = nameLengths[i];
        sum = sum + age;
    }
    System.out.println("6 Answer: " + sum);

}
```

Challenge 7:

```java
/**
 * stringIntConcat
 * Takes a String, word, and an integer, n, as arguments and
 * returns the word concatenated to itself n number of times.
 *
 * @param String word
 * @param int    n
 * @return String concatWords
 */
private static String stringIntConcat(String word, int n) {

    String concatWords = "";
    for (int i = 1; i <= n; i++) {
        concatWords = concatWords.concat(word);
    }
    System.out.println("7 Answer: " + concatWords + " - Returning the concatenated words.");
    return concatWords;

}
```

Challenge 8:

```java
/**
 * namingStrings
 * Takes two Strings, firstName and lastName, and returns a full
 * name separated by a space.
 *
 * @param String firstName
 * @param String lastName
 * @return String fullName
 */
private static String namingStrings(String firstName, String lastName) {

    String fullName = firstName + " " + lastName;
    System.out.println("8 Answer: " + fullName);
    return fullName;

}
```

Challenge 9:

```java
/**
 * sumOfIntArray
 * Takes and array of int and returns true if the sum of all the
 * ints in the array is greater than 100. Assumes the value is NOT equal to 100
 *
 * @param int[] intAry
 * @return boolean
 */
private static boolean sumOfIntArray(int[] intAry) {

    int total = 0;
    for (int i = 0; i < intAry.length; i++) {
        int element = intAry[i];
        total = total + element;
    }

    if (total > 100) {
        System.out.println("9 Answer: The total " + total + " is greater than 100. Return true.");
        return true;
    }
    System.out.println("9 Answer: The total " + total + " is less than 100. Return false.");
    return false;
}
```

Challenge 10:

```java
/**
 * avgOfDoubleArray
 * Takes an array of double and returns the average of all the
 * elements in the array.
 *
 * @param double[] testDblAry
 * @return double avg
 */
private static double avgOfDoubleArray(double[] dblAry) {

    double sum = 0.00;
    for (int i = 0; i < dblAry.length; i++) {
        double element = dblAry[i];
        sum = sum + element;
    }
    double avg = sum / dblAry.length;
    System.out.println("10 Answer: The average with no rounding added to the answer is: " + avg
            + " . Returning this number as the average.");

    return avg;

}
```

Challenge 11:

```java
/**
 * doubleTroubleArrays
 * Takes two arrays of double Returns true if the average of
 * the elements in the first array is greater than the average of the elements
 * in the second array Assumes the two values are NOT equal
 *
 * @param double[] dblAry1
 * @param double[] dblAry2
 * @return boolean
 */
private static boolean doubleTroubleArrays(double[] dblAry1, double[] dblAry2) {

    // Average of dblAry1
    double sum1 = 0.00;
    for (int i = 0; i < dblAry1.length; i++) {
        double element1 = dblAry1[i];
        sum1 = sum1 + element1;
    }
    double avg1 = sum1 / dblAry1.length;

    // Average of dblAry2
    double sum2 = 0.00;
    for (int i = 0; i < dblAry2.length; i++) {
        double element2 = dblAry2[i];
        sum2 = sum2 + element2;
    }
    double avg2 = sum2 / dblAry2.length;

    // Compare the averages.
    if (avg1 > avg2) {
        System.out.println("11 Answer: The average for the first array " + avg1 + " is greater than the second "
                + avg2 + ". Returning true.");
        return true;
    }

    System.out.println("11 Answer: The average for the second array " + avg2 + " is greater than the first " + avg1
            + ". Returning false.");
    return false;
}
```

Challenge 12:

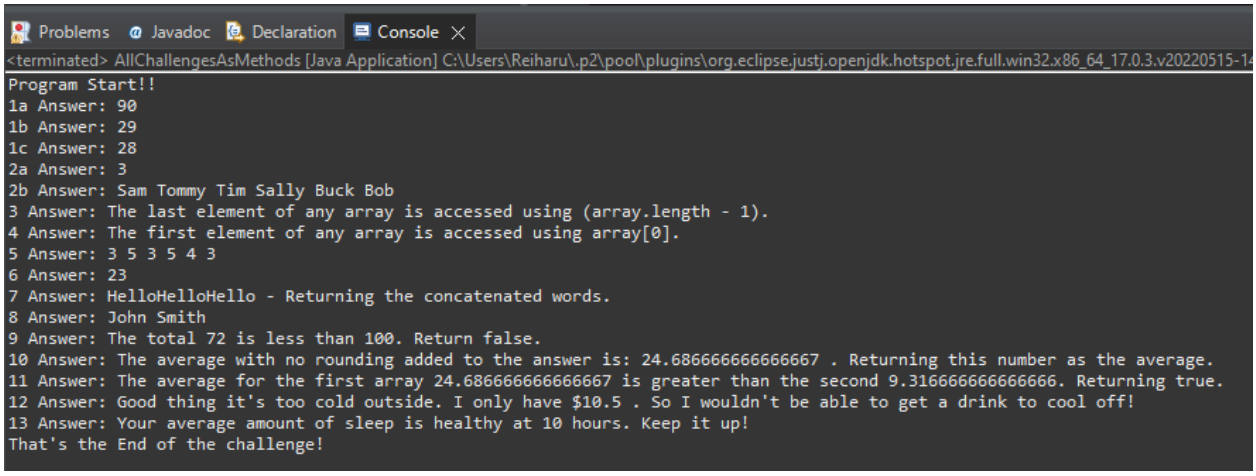
```java
/**
 * willBuyDrink
 * Takes a boolean isHotOutside, and a double moneyInPocket.
 * Returns true if it is hot outside and if moneyInPocket is greater than 10.50.
 *
 * @param isHotOutside
 * @param moneyInPocket
 * @return boolean
 */
private static boolean willBuyDrink(boolean isHotOutside, double moneyInPocket) {

    if (isHotOutside && moneyInPocket > 10.50) {
        System.out.println("12 Answer: It's too hot outside. Buying a cold drink for: $" + moneyInPocket);
        return true;
    } else if (isHotOutside && moneyInPocket <= 10.50) {
        System.out.println("12 Answer: It's too hot outside. But I only have $" + moneyInPocket
                + " . I can't get a drink to cool off!");
        return false;
    } else if (!isHotOutside && moneyInPocket > 10.50) {
        System.out.println("12 Answer: It's too cold outside. I'll keep $" + moneyInPocket + " in my pocket.");
        return false;
    }

    System.out.println("12 Answer: Good thing it's too cold outside. I only have $" + moneyInPocket
            + " . So I wouldn't be able to get a drink to cool off!");
    return false;
}
```

Challenge 13:

```java
/**
 * properNightsRest
 * This method checks the average of the total hours slept per
 * night and tells the user whether or not they need to get more rest or less
 *
 * @param int[] goToBed
 * @param int[] wakeUp
 */
private static void properNightsRest(int[] goToBed, int[] wakeUp) {

    if (goToBed.length != wakeUp.length) {
        System.out.println("13 Answer: One of your arrays is longer than the other! Exiting.");
    } else {

        // Finds the hours slept per night
        int[] hoursSlept = new int[goToBed.length];
        for (int i = 0; i < goToBed.length; i++) {
            hoursSlept[i] = Math.abs(goToBed[i] - wakeUp[i]);
        }

        // Finds the average hours slept within the number of days specified by the
        // lengths of the arrays.
        int total = 0;
        for (int i = 0; i < hoursSlept.length; i++) {
            int element = hoursSlept[i];
            total = total + element;
        }

        int avgHrsPerWk = total / hoursSlept.length;

        if (avgHrsPerWk >= 8) {
            System.out.println(
                    "13 Answer: Your average amount of sleep is healthy at " + avgHrsPerWk + " hours. Keep it up!");
        } else {
            System.out.println("13 Answer: Your average amount of sleep is unhealthy at " + avgHrsPerWk
                    + " hours. You need more sleep!");
        }
    }

}
```

Screenshots of Running Application:

All methods ran at the same time to make things easier.



URL to GitHub Repository:

https://github.com/KitiaraJ/BootCamp/tree/main/java-wk3-challenge