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
public class CodingProjectW03 {

    public static void main(String[] args) {

        int[] ages = {3, 9, 23, 64, 2, 8, 28, 93};

        //1)a. Subtract first age from last age without using ages[7]
        System.out.println((ages[ages.length - 1]) - ages[0]);

        //1)b. Add a new age to the array and repeat step above to ensure dynamic
            //added age "34" to the array and confirmed dynamic

        //1)c. Use a  loop to calculate and print the average age
        int sum = 0;
        for (int age : ages) {
        sum += age;
        }

        int avg = sum / ages.length;
        System.out.println(avg);
```

Problems Javadoc Declaration Console

```
<terminated> CodingProjectW03 [Java Application] C:\Program Files\Java\jdk-11.0.15\bin\javaw.exe (Jul 14, 2022, 2:29:06 PM – 2:29:06 PM) [pid: 13616]
90
28
```

==========

```java
//#2. Create array of String called names
        String[] names = {"Sam", "Tommy", "Sally", "Buck", "Bob"};

    //2)a. Use a loop to calculate and print the average number of letters per name
        int totalLetters = 0;

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

        int avgLetters = totalLetters / names.length;
        System.out.println("#2/a: ");
        System.out.println(avgLetters);

    //2)b. Use a loop to concatenate all the names together, separated by spaces, print
        System.out.println("#2/b: ");

        for (int i = 0; i < names.length; i++) {
            System.out.print((names[i]) + " ");
        }
```

Problems Javadoc Declaration Console

```
<terminated> CodingProjectW03 [Java Application] C:\Program Files\Java\jdk-11.0.15\bin\javaw.exe (Jul 16, 2022, 12:46:23 PM – 12:46:24 PM) [pid: 18360]
#2/a:
4
#2/b:
Sam Tommy Sally Buck Bob
```

```java
//#3 How do you access the last element of any array?
            // array[array.length - 1];

//#4 How do you access the first element of any array?
            // array[0];

//#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.
        //System.out.println("#5: ");
        int[] nameLengths = new int [names.length];

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

//#6 Write a loop to iterate over the nameLengths array & calculate the sum of all the elements in the array. Print resu
        int sumLengths = 0;   //start at value zero because thats where you start counting from
        for (int num : nameLengths) {
        sumLengths += num;
        }
        System.out.println("#6: ");
        System.out.println(sumLengths);
```

Problems Javadoc Declaration Console

```
<terminated> CodingProjectW03 [Java Application] C:\Program Files\Java\jdk-11.0.15\bin\javaw.exe (Jul 16, 2022, 12:54:07 PM – 12:54:07 PM) [pid: 5092]
3
5
5
4
3
#6:
20
```

==========

```java
//#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. "Hello" and 3, would return "HelloHelloHello").

    public static String multiplyString(String word, int n) {
        String wordTimesN = "";
        for (int i = 0; i< n; i++) {
            wordTimesN += word;
        }
        return wordTimesN;
    }

//#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).

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


//#9 Write a method that takes an array of int and returns true if the sum of all the ints in the array is > 100.

    public static boolean isSumOver100(int[] ages) {
        int sum = 0;
        for (int age : ages) {
            sum += age;
        }
        return (sum > 100);
    }

//#10 Write a method that takes an array of double and returns the average of all the elements in the array.

    public static double avgOfDoubles (double[] doubleArray) {
        double sumDub = 0;
        double avgDub = 0;
        for (double dub : doubleArray) {
            sumDub += dub;
            avgDub = sumDub / doubleArray.length;
        }
        return avgDub;
    }
```

```java
//#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.

    public static boolean avg1GreaterThanAvg2(double[] doubleArray1, double[] doubleArray2) {
        double sum1 = 0;
        double sum2 = 0;
        double avg1 = 0;
        double avg2 = 0;

        for (double dub1 : doubleArray1) {
            sum1 += dub1;
            avg1 = sum1 / doubleArray1.length;
        }
        for (double dub2 : doubleArray2) {
            sum2 += dub2;
            avg2 = sum2 / doubleArray2.length;
        }
        return sum1 > sum2;
    }

//#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.

    public static boolean willBuyDrink (boolean isHotOutside, double moneyInPocket) {
        return ((isHotOutside == true) && (moneyInPocket>10.50));
}
```

==========

```java
//#13 Create a method of your own that solves a problem. In comments, write what the method
    //does and why you created it.

    public static boolean needHearingProtection (double soundLevel, double hours) {
        if(soundLevel <=90.0 && hours <=8.0) {
            return true;
            } else if (soundLevel <=92.0 && hours <=6.0) {
                return true;
            } else if (soundLevel <=95.0 && hours <=4.0) {
                return true;
            } else if (soundLevel <=97.0 && hours <=3.0) {
                return true;
            } else if (soundLevel <=100.0 && hours <=2.0) {
                return true;
            } else if (soundLevel <=102.0 && hours <=1.5) {
                return true;
            } else if (soundLevel <=105.0 && hours <=1.0) {
                return true;
            } else if (soundLevel <=110.0 && hours <=0.5) {
                return true;
            } else if (soundLevel <=115.0 && hours <=0.25) {
                return true;
            } else {
                return false;
            }
    }

    //I currently work as a Safety Manager, so I decided to create a method that determines whether an employee can
    //work in the given working conditions, with the input being the sound level (in dBa) and the anticipated hours
    //they plan to work. A return of TRUE means they are safe to work under those conditions. A return of FALSE means
    //the work conditions are not safe and they need to adjust the work conditions, whether it is lowering the noise
    //level or the hours to be worked.
```

URL to GitHub Repository:

https://github.com/JaxYoungblood/Week03CodingHW.git