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.

```java
// Part 1.
int ages[] = { 9, 23, 64, 2, 8, 28, 93 }; // creation of age array

int lastElementIndex = ages.length;
System.out.println("The last element of the array, minus the first element equals: "
        + (ages[lastElementIndex - 1] - ages[0])); // Part a

double sum = 0;
for (int numbers : ages) {
    sum += numbers;
}
System.out.println("The avarage of the array is: " + String.format("%.2f", (sum / lastElementIndex))); // Part c
```

```
The last element of the array, minus the first element equals: 84
The avarage of the array is: 32.43
```

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.

```java
//  Part 2. a

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

        for (String letters : names) {
            index += letters.length();

        }

        System.out.println("The avarage number of letters in the array is: " + index / names.length);

        // Part b, Concatenate elements of the array

        String sum2 = "";

        for (String name : names) {
            sum2 += (name + " ");
        }
        System.out.println(sum2);
```

```
The avarage number of letters in the array is: 3
Sam Tommy Tim Sally Buck Bob
```

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.

```java
//Part 5
        int nameLengths[] = new int[names.length];
        int j = 0;
        int k = 0;
        for (String letters : names) {
            j = letters.length();
            nameLengths[k] = j;
            k++;
        }

        System.out.println(Arrays.toString(nameLengths));
```

```
[3, 5, 3, 5, 4, 3]
```

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.

```java
//Part 6
        int sum5 = 0;

        for (int count : nameLengths) {
            sum5 += count;

        }
        System.out.println("Sum of all the elements in the array is: " + sum5);
```

```
Sum of all the elements in the array is: 23
```

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”).

```java
// Part 7
    public static String myword(String word, int n) {
        String answer = "";
        for (int i = 0; i < n; i++) {
            answer += word;
        }

        return answer;
    }
```

```java
//Part 7 Method Return
        System.out.println(myword("Hello", 3));
```

```
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).

```java
//Part 8
    public static String myName(String name1, String name2) {

        String answer = name1 + " " + name2;

        return answer;
    }
```

```java
//Part 8 Method Return
       System.out.println(myName("Jeus", "Romero"));
```

```
Jeus Romero
```

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.

```java
//Part 9
    public static boolean myArray(int[] num) {

        int sum = 0;
        for (int numbers : num) {

            sum += numbers;

        }

        if (sum >= 100) {

            return true;
        } else
            return false;

    }
```

```java
// Part 9 Method Return
        int testArray[] = { 50, 50 };
        System.out.println(myArray(testArray));
```

```
true
```

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

```java
//Part 10

    public static double myDouble(double[] num) {

        double sum = 0;
        for (double numbers : num) {

            sum += numbers;

        }
        return sum / num.length;
    }
```

```java
//Part 10
        double doubleArray[] = { 10.00, 15.5, 20.5 };
        System.out.println(String.format("%.2f", myDouble(doubleArray)));
```

```
15.33
```

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.

```java
//Part 11
    public static boolean avg(double[] x, double[] y) {
        if (myDouble(x) > myDouble(y)) {
            return true;
        } else
            return false;

    }
```

```java
//Part 11 Method Return
        double avgArray[] = { 50.00, 50.00 };
        System.out.println(avg(avgArray, doubleArray));
```

```
true
```

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.

```java
//Part 12
    public static boolean willBuyDrink(boolean isHotOUtside, double moneyInPocket) {

        if (isHotOUtside && moneyInPocket > 10.50)
            return true;
        else
            return false;
    }
```

```
//Part 12 Method return
        System.out.println(willBuyDrink(true, 10.51));
```

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

findEvenOdd will return weather a int is odd or even.

Screenshots of Code:

```
//Part 13
    public static void findEvenOdd(int num)  // is an int odd or even.
    {
    //method body
    if(num%2==0)
    System.out.println(num+" is even");
    else
    System.out.println(num+" is odd");
    }
```

Screenshots of Running Application:

```
//Part 13 Method
        findEvenOdd(256);
```

```
256 is even
```

URL to GitHub Repository:

[File Finder (github.com)]