Intro to Java Week 3 Coding Assignment

Points possible: 70

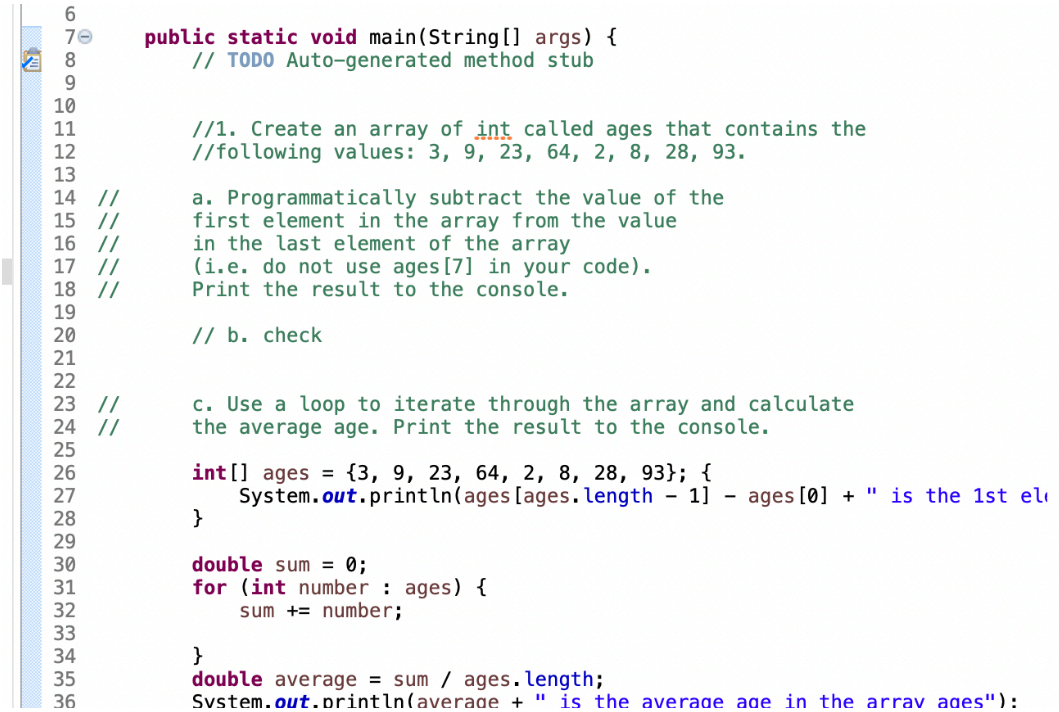
Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity 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:

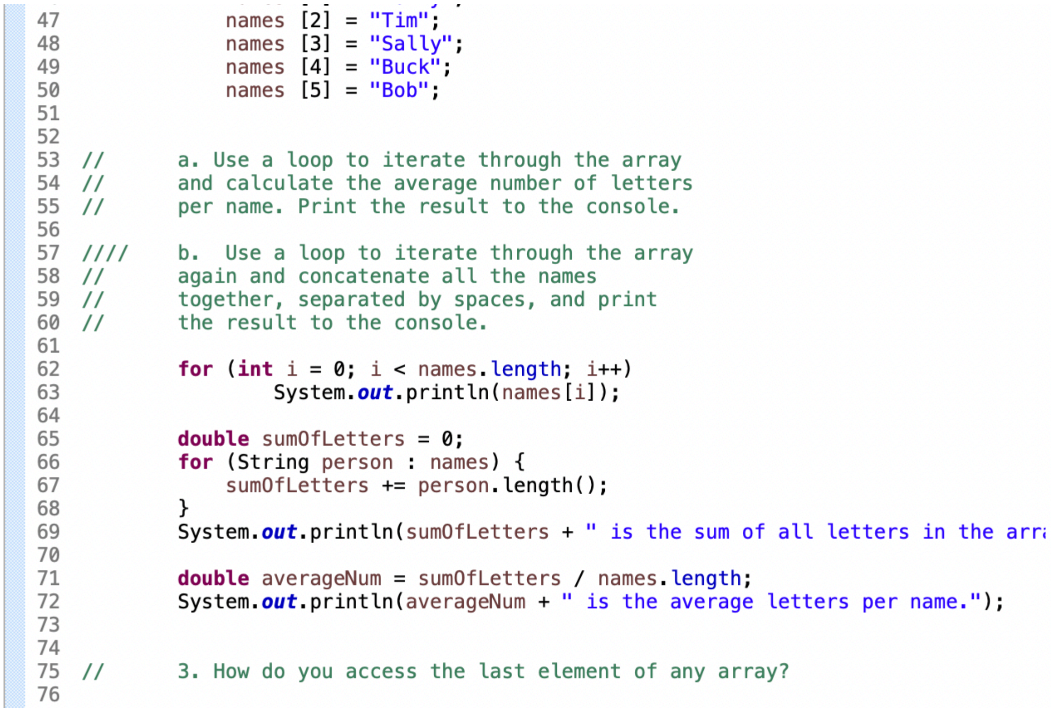
1. Create an array of int called ages that contains the following values: 3, 9, 23, 64, 2, 8, 28, 93.
    a. Programmatically subtract the value of the first element in the array from the value in the last element of the array (i.e. do not use ages[7] in your code). Print the result to the console.
    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.



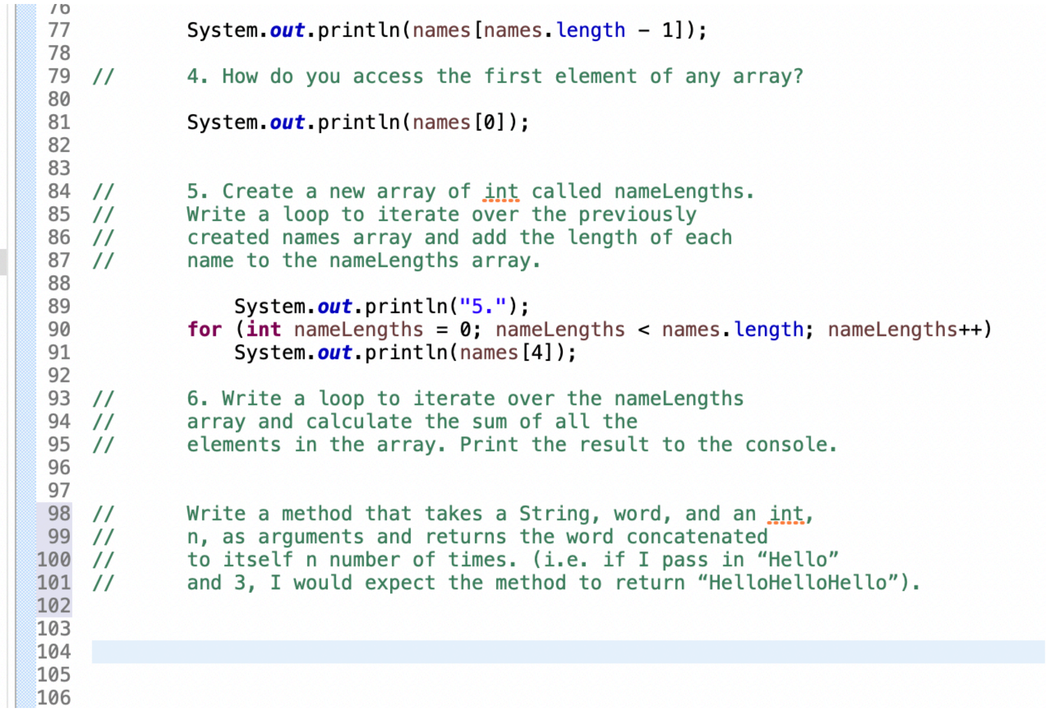
```

    public static void main(String[] args) {
        // TODO Auto-generated method stub


        //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. check


//      c. Use a loop to iterate through the array and calculate
//      the average age. Print the result to the console.

        int[] ages = {3, 9, 23, 64, 2, 8, 28, 93}; {
            System.out.println(ages[ages.length - 1] - ages[0] + " is the 1st el
        }

        double sum = 0;
        for (int number : ages) {
            sum += number;

        }
        double average = sum / ages.length;
        System.out.println(average + " is the average age in the array ages");
```

1/3 code



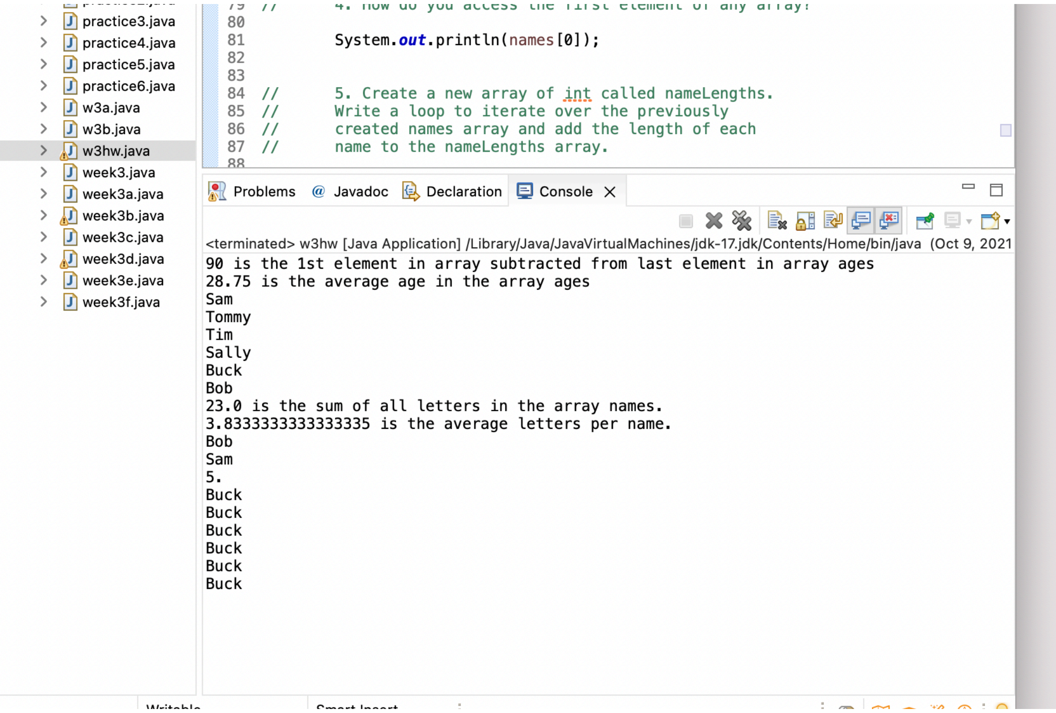
```
            names [2] = "Tim";
            names [3] = "Sally";
            names [4] = "Buck";
            names [5] = "Bob";


//      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.

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

        double sumOfLetters = 0;
        for (String person : names) {
            sumOfLetters += person.length();
        }
        System.out.println(sumOfLetters + " is the sum of all letters in the arr

        double averageNum = sumOfLetters / names.length;
        System.out.println(averageNum + " is the average letters per name.");


//      3. How do you access the last element of any array?

```

2/3 code



```
        System.out.println(names[names.length - 1]);

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

        System.out.println(names[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.");
        for (int nameLengths = 0; nameLengths < names.length; nameLengths++)
            System.out.println(names[4]);

//      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.


//      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”).
```

3/3 code



```
        System.out.println(names[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.
```

```
<terminated> w3hw [Java Application] /Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home/bin/java  (Oct 9, 2021
90 is the 1st element in array subtracted from last element in array ages
28.75 is the average age in the array ages
Sam
Tommy
Tim
Sally
Buck
Bob
23.0 is the sum of all letters in the array names.
3.8333333333333335 is the average letters per name.
Bob
Sam
5.
Buck
Buck
Buck
Buck
Buck
Buck
```

Output