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?
   myArray[myArray.length-1]
4. How do you access the first element of any array?
   myArray[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.
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.

I made a method to convert Fahrenheit to Celsius, I created it to convert units of temperate easily.

Screenshots of Code:

```java
public class PromineoTechWeek3 {

    //Variables used in methods to showcase usage

    //Step 1 variables
    int numbers[] = {3, 9, 23, 64, 2, 8, 28, 93};
    //Used in step 1.a
    int dynamicNumbers[] = {3, 9, 23, 64, 2, 8, 28, 93, 18};

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

    //Step 5 variables
    int nameLengths[];

    /**Step 1
     * This method will return the difference of first and last number in array
     *
     * @return - an int number for the difference
     */
    public int subFirstLast() {
        if(numbers.length > 0) {
            return numbers[numbers.length-1]-numbers[0];
        }else {
            return 0;
        }

    }



    /**Step 1, ensuring it is dynamic, calculates from an array of a different size
     * This method will return the difference of first and last number in array
     *
     * @return - an int number for the difference
     */
    public int subFirstLastDynamic() {
        if(dynamicNumbers.length > 0) {
            return dynamicNumbers[dynamicNumbers.length-1]-dynamicNumbers[0];
        }else {
            return 0;
        }

    }


    /**Step 1.c, use a loop to find the average age.
     * This method will iterate through an array to find the average of an array
     *
     * @return - a float for the average age of the array
     */
    public float arrayAverage() {
```

```java
    /**Step 1.c, use a loop to find the average age.
     * This method will iterate through an array to find the average of an array
     *
     * @return - a float for the average age of the array
     */
    public float arrayAverage() {
        float sum = 0;
        //Check for no numbers case
        if(numbers.length > 0) {
            for(int i : numbers) {
                sum += i;
            }
            return sum/numbers.length;
        }else {
            return 0;
        }
    }


    /**Step 2.a
     * Use a loop to iterate and count characters of elements in the array
     *
     * @return - a float number for the average amount of characters
     */
    public float avgNameLength() {
        float sumChars = 0;
        for(String s : names) {
            sumChars += s.length();
        }
        return sumChars/names.length;
    }


    /**Step 2.b
     * This method with iterate through an array of strings and return it concatenated
     *
     * @return myStr - a pre-formatted string for the concatenated names
     */
    public String printConcatNames() {
        String myStr = "";
        for(String s : names) {
            myStr += s;
            myStr += " ";
        }
        return myStr;
    }


    /**Step 5
     * modifies the value of the int array, 'nameLengths'.
     */
    public void nameLengthArray() {
```

```java
    /**Step 5
     * modifies the value of the int array, 'nameLengths'.
     */
    public void nameLengthArray() {
        int tempArray[] = new int[names.length];
        for(int i = 0; i < names.length ; i++) {
            tempArray[i] = names[i].length();
        }
        nameLengths = tempArray;
    }


    /**Method used for printing an int array
     *
     * @param myArray - an integer array to print
     */
    public void printArray(int[] myArray) {
        String formatPrintArray = "[";
        for(int i = 0; i < myArray.length-1; i++) {
            formatPrintArray += myArray[i];
            formatPrintArray += ", ";
        }
        formatPrintArray += myArray[myArray.length-1] + "]";
        System.out.println(formatPrintArray);
    }


    /**Step 6
     * This method iterates over an int array and prints the sum value
     * @param myArray - an integer array
     * @return sum - the sum of values in myArray
     */
    public int countArraySum(int[] myArray) {
        int sum = 0;
        for(int i: myArray) {
            sum += i;
        }
        return sum;
    }


    /**Step 7
     * This method takes two parameters for a string and an int, and repeats the string int many times
     *
     * @param word - A string to repeat
     * @param n - number of times to repeat
     * @return outputString - the resulting repeated string
     */
    public String wordCopier(String word, int n) {
```

```java
    /**Step 7
     * This method takes two parameters for a string and an int, and repeats the string int many times
     *
     * @param word - A string to repeat
     * @param n - number of times to repeat
     * @return outputString - the resulting repeated string
     */
    public String wordCopier(String word, int n) {
        String outputString = "";
        for(int i = 0; i<n; i++) {
            outputString += word;
        }
        return outputString;
    }

    /**Step 8
     * This method takes two string parameters and returns a concatenated String
     *
     * @param firstName - a string parameter for the first name
     * @param lastName  - a string paramater for the last name
     * @return - a String combination of the first and last name
     */
    public String nameConcat(String firstName, String lastName) {
        return firstName + " " + lastName;
    }


    /**Step 9
     * This method takes myNum, an int array as a parameter and checks the sum of elements to be greater than 100
     *
     * @param myNum - an integer array for be summed then compared
     * @return - a boolean value true if the value is greater than 100, otherwise false
     */
    public boolean greaterThanHundred(int[] myNum) {
        int sum = 0;
        for(int i : myNum) {
            sum += i;
        }
        if(sum > 100) {
            return true;
        }else {
            return false;
        }
    }

    /**Step 10
     * This method takes doubleArray, a double array as a parameter and returns the average value of its elements
     *
     * @param doubleArray - the array to find the average of
     * @return - the average value of doubleArray elements
     */
    public double doublesAvg(double doubleArray[]) {
```

```java
    /**Step 10
     * This method takes doubleArray, a double array as a parameter and returns the average value of its elements
     *
     * @param doubleArray - the array to find the average of
     * @return - the average value of doubleArray elements
     */
    public double doublesAvg(double doubleArray[]) {
        double doubleAverage = 0;
        for(double d : doubleArray) {
            doubleAverage += d;
        }
        //Divide by zero prevention
        if(doubleArray.length > 0) {
            return doubleAverage/doubleArray.length;
        }else {
            return 0;
        }

    }


    /**Step 11
     * This method takes two double arrays and returns true if the average of the first array is greater than the second
     *
     *
     * @param firstArray - the first array in the comparison. returns true if larger second
     * @param secondArray - the second array in the comparison. returns false if larger than first
     * @return -  a boolean true if the average of the first array is greater than the second, otherwise returns false
     */
    public boolean doubleArrayCompare(double firstArray[], double secondArray[]) {
        if(this.doublesAvg(firstArray) > this.doublesAvg(secondArray)) {
            return true;
        }else {
            return false;
        }
    }


    /**Step 12
     * This method takes two parameters isHotOutside and moneyInPocket and
     *
     * @param isHotOutside - a boolean value of true if it is hot outside, false if not
     * @param moneyInPocket - a double value of money in pocket
     * @return - a boolean true if isHotOutside is true, and moneyInPocket is greater than 10.50, else false
     */
    public boolean willBuyDrink(boolean isHotOutside, double moneyInPocket) {
```

```java
     /**Step 12
      * This method takes two parameters isHotOutside and moneyInPocket and
      *
      * @param isHotOutside - a boolean value of true if it is hot outside, false if not
      * @param moneyInPocket - a double value of money in pocket
      * @return - a boolean true if isHotOutside is true, and moneyInPocket is greater than 10.50, else false
      */
     public boolean willBuyDrink(boolean isHotOutside, double moneyInPocket) {
         if(moneyInPocket > 10.50 && isHotOutside) {
             return true;
         }else {
             return false;
         }
     }

     /**Step 13
      * A method to convert fahrenheit units into celcius units
      * takes in a fahrenheit temperature and returns its celcius conversion
      *
      * @param temperature - the double value of fahrenheit temperature
      * @return - the double value of its temperature in celcius
      */
     public double fahrenheitConvert(double temperature) {
         return (temperature - 32) * 5 / 9;
     }

     public static void main(String[] args) {
```

```
245     public static void main(String[] args) {
246         PromineoTechWeek3 myObj = new PromineoTechWeek3();
247         //Step 1
248         System.out.println(myObj.subFirstLast());
249 
250         //Step 1.b
251         System.out.println(myObj.subFirstLastDynamic());
252 
253         //Step 1.c
254         System.out.println(myObj.arrayAverage());
255 
256         //Step 2.a
257         System.out.println(myObj.avgNameLength());
258 
259         //Step 2.b
260         System.out.println(myObj.printConcatNames());
261 
262         //Step 5
263         myObj.nameLengthArray();
264         //Prints step 5
265         myObj.printArray(myObj.nameLengths);
266 
267         //Step 6
268         System.out.println(myObj.countArraySum(myObj.nameLengths));
269 
270         //Step 7
271         System.out.println(myObj.wordCopier("Hello", 3));
272 
273         //Step 9
274         System.out.println(myObj.nameConcat("John", "Smith"));
275 
276         //Step 9
277         System.out.println(myObj.greaterThanHundred(myObj.nameLengths));
278 
279         //Step 10
280         double stepTenArray[] = {1.2, 2.3, 3.4, 100.34};
281         System.out.println(myObj.doublesAvg(stepTenArray));
282 
283         //Step 11
284         double stepElevenArray[] = {23.5, 234.1, 123.3};
285         System.out.println(myObj.doubleArrayCompare(stepElevenArray, stepTenArray));
286         System.out.println(myObj.doubleArrayCompare(stepTenArray, stepElevenArray));
287 
288         //Step 12
289         boolean hotDay = true;
290         double pocketChange = 20;
291         System.out.println(myObj.willBuyDrink(hotDay, pocketChange));
292 
293         //Step 13
294         double fahrenheitTemp = 350;
295         System.out.println(myObj.fahrenheitConvert(fahrenheitTemp));
296 
297     }
```

Screenshots of Running Application:

```
90
15
28.75
3.8333333
Sam Tommy Tim Sally Buck Bob
[3, 5, 3, 5, 4, 3]
23
HelloHelloHello
John Smith
false
26.810000000000002
true
false
true
176.66666666666666
```

URL to GitHub Repository:

https://github.com/rainphantasm/PromineoWeek3/blob/main/PromineoTechWeek3.java