Intro to Java Week 3 Coding Assignment

Lucas Veenendaal

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:

```
package Week3Assignment;

public class Week3 {

    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.
            //Programmatically subtract the value of the first element in the array from the value in the last element of the array.
            //Print the result to the console.

            int[] ages = {3, 9, 23, 64, 2, 8, 28, 93, 73};

            int firstFromLast = ages[ages.length - 1] - ages[0];

                    System.out.println("Question 1-A:");
                        System.out.println(firstFromLast); //outputs 90 (93-3) before 73 was added in part 2

            // Add a new age to your array and repeat the step above to ensure it is dynamic (works for arrays of different lengths).

                    System.out.println("Question 1-B:");
                        System.out.println(firstFromLast); //outputs 70 (73 - 3)

            //Use a loop to iterate through the array and calculate the average age. Print the result to the console.
            double averageAges = 0;

            for (int i = 0; i < ages.length; i++) {

                averageAges += ages[i];
            }

            averageAges /= ages.length;

                    System.out.println("Question 1-C:");
                        System.out.println(averageAges);

```

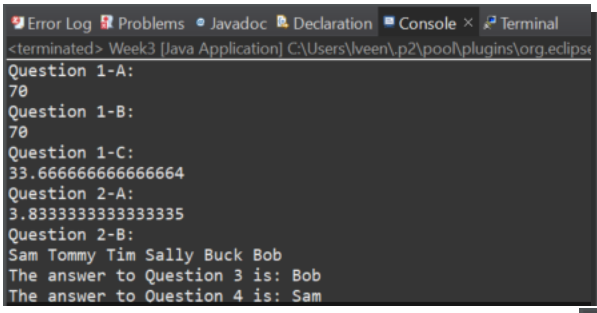
```java
        //2
            //Create an array of String called names that contains the following values: "Sam", "Tommy", "Tim", "Sally", "Buck", "Bob".
            //Use a loop to iterate through the array and calculate the average number of letters per name.
            //Print the result to the console.

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

            double averageNameLength = 0;

            for (int i = 0; i < names.length; i++) {

                averageNameLength += names[i].length();
            }

            averageNameLength /= names.length;

                    System.out.println("Question 2-A:");
                        System.out.println(averageNameLength);

            //Use a loop to iterate through the array again and concatenate all the names together, separated by spaces.
            //Print the result to the console.

            String namesConcatenated = "";

            for (int i = 0; i < names.length; i++) {

                namesConcatenated += (names[i] + " ");
            }

                    System.out.println("Question 2-B:");
                        System.out.println(namesConcatenated);

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

                String answerNumberThree = "The answer to Question 3 is: " + names[names.length - 1]; //Will work with any length of array.

                    System.out.println(answerNumberThree);


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

                String answerNumberFour = "The answer to Question 4 is: " + names[0]; //Always 0 since array elements are labeled beginning with base 0.

                    System.out.println(answerNumberFour);

        //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.

                int[] nameLengths = new int[6];

                for(int i = 0; i < nameLengths.length; i++) {

                    nameLengths[i] = names[i].length();

                }

        //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.

                int nameLengthsSum = 0;

                for(int i = 0; i < nameLengths.length; i++) {

                    nameLengthsSum += nameLengths[i];

                }

                    System.out.println("Question 6:");
                        System.out.println(nameLengthsSum);

        //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").

                    System.out.println("Question 7 Test:");
                        System.out.println(concatenator("Hello",3));


        //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).

                    System.out.println("Question 8 Test:");
                        System.out.println(fullName("Bobby", "Jones"));

        //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.

                int[] hundredArrayTest = { 40, 30, 20, 11 };
                int[] hundredArrayTest2 = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

                    System.out.println("Question 9 Tests:");
                        System.out.println(hundredArray(hundredArrayTest));     // Sum of 101, returns true.
                        System.out.println(hundredArray(hundredArrayTest2));    // Sum of 45, returns false.

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

                double [] avDoubleTest = { 1.2, 2.2, 3.2, 4.4 }; // Sum of 11, average of 2.75.
                double [] avDoubleTest2 = {10.5, 23.7, 14.3, 6.2, 37.8}; // Sum of 92, average of 18.4.

                    System.out.println("Question 10 Tests:");
                        System.out.println(averageDouble(avDoubleTest)); //returns the average of 2.75.
                        System.out.println(averageDouble(avDoubleTest2)); //returns the average of 18.4.

        //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.
                    System.out.println("Question 11 Tests:");
                        System.out.println(compareDoubles(avDoubleTest,avDoubleTest2)); //See Q-10: avDoubleTest = 2.75, avDoubleTest2 = 18.4  returns false since 2.75 is not greater than 18.4
                        System.out.println(compareDoubles(avDoubleTest2, avDoubleTest)); // returns true because the inverse of the above line is 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.

                    System.out.println("Question 12 Tests:");
                        System.out.println(willBuyDrink(true,11.5)); //It is hot out, and you have more than 10.5, so it returns true.
                        System.out.println(willBuyDrink(false,11.5)); //It is not hot out, so it returns false even though you have enough money.
                        System.out.println(willBuyDrink(true,9.0)); //It is hot out, but you don't have enough for a drink, so it returns false
                        System.out.println(willBuyDrink(false,9.0)); //It is not hot out, and you couldn't afford a drink anyway, so it returns false.
```

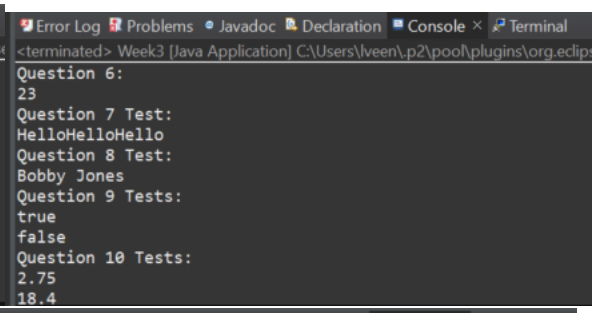
```
159
160         //13
161             //Create a method of your own that solves a problem. In comments, write what the method does and why you created it.
162                 //My custom method will be called willBuyGas which takes a boolean lowOnGas, a boolean drivingTomorrow, and a double accountBalance.
163                 //If you are going somewhere tomorrow, low on gas, and have enough in your account, it will return "you will buy gas."
164                 //If you are going somewhere tomorrow, low on gas, but don't have enough money, it returns "Can't afford Gas."
165                 //If you are going somewhere tomorrow, but are not low on gas, it returns "You have enough gas for tomorrow."
166                 //If you are not going anywhere tomorrow, it returns "You don't need to worry about gas until tomorrow."
167
168                     System.out.println("Question 13 Tests:");
169                         willBuyGas(true,true,75.23); //going somewhere, need gas, can afford gas
170                         willBuyGas(true,true,23.75); //going somewhere, need gas, can't afford gas
171                         willBuyGas(true,false,23.75); //going somewhere, don't need gas, can't afford gas
172                         willBuyGas(false,false,23.75); //not going somewhere, don't need gas, can't afford gas
173
174
175     }
176     //Methods
177         //7
178         public static String concatenator (String word, int n) {
179                 String stringConcatenated = "";
180
181                 for (int i = 0; i < n; i++) {
182                     stringConcatenated += word;
183                 }
184                     return stringConcatenated;
185
186             }
187
188         //8
189         public static String fullName (String firstName, String lastName) {
190
191             String fullName = firstName + " " + lastName;
192
193             return fullName;
194
195             }
196
```

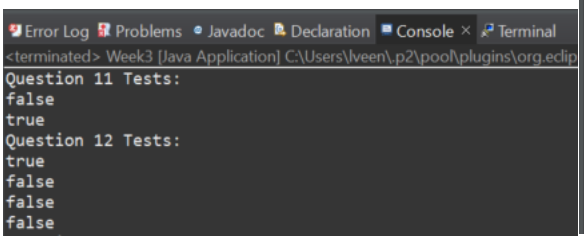
```
197         //9
198         public static boolean hundredArray (int[] arr) {
199
200             boolean greaterThanHundred;
201             int arrSum = 0;
202
203             for (int i = 0; i < arr.length; i++) {
204
205                  arrSum += arr[i];
206
207             }
208
209             if ( arrSum > 100) {
210                 greaterThanHundred = true;
211             } else {
212                 greaterThanHundred = false;
213             }
214
215             return greaterThanHundred;
216         }
217
218         //10
219         public static double averageDouble (double[] arr1) {
220
221             double arr1Sum = 0;
222
223             for (int i = 0; i < arr1.length; i++) {
224                 arr1Sum += arr1[i];
225             }
226                 arr1Sum /= arr1.length;
227
228                 return arr1Sum;
229
230             }
231
```

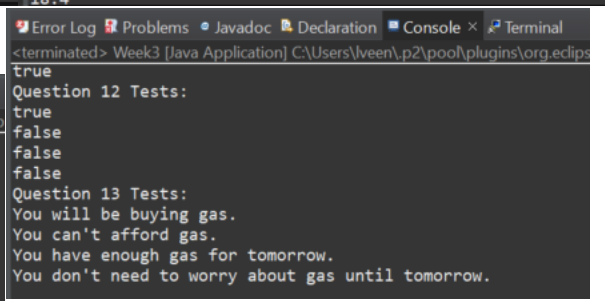
```
232         //11
233         public static boolean compareDoubles (double[] dub1, double[] dub2) {
234
235             boolean firstDoubleGreater;
236
237             if (averageDouble(dub1) > averageDouble(dub2)) {
238                 firstDoubleGreater = true;
239             } else {
240                 firstDoubleGreater = false;
241             }
242                 return firstDoubleGreater;
243
244             }
245
246         //12
247         public static boolean willBuyDrink (boolean isHotOutside, double moneyInPocket) {
248
249             boolean makePurchase;
250
251             if(isHotOutside == true && moneyInPocket > 10.5) {
252                 makePurchase = true;
253             } else {
254                 makePurchase = false;
255             }
256                 return makePurchase;
257
258             }
259
260         //13
261         public static void willBuyGas (boolean drivingTomorrow, boolean lowOnGas, double accountBalance) {
262
263             if(drivingTomorrow == true && lowOnGas == true && accountBalance > 50.00) {
264                 System.out.println("You will be buying gas.");
265             } else if (drivingTomorrow == true && lowOnGas == true && accountBalance < 50.00) {
266                 System.out.println("You can't afford gas.");
267             } else if (drivingTomorrow == true && lowOnGas == false) {
268                 System.out.println("You have enough gas for tomorrow.");
269             } else if (drivingTomorrow == false) {
270                 System.out.println("You don't need to worry about gas until tomorrow.");
271             }
272         }
273
274 }
275
```

Screenshots of Running Application:









URL to GitHub Repository:

https://github.com/LVeenendaal/Week-3