PROMINEO TECH

Intro to Java Weeks 3-4 Coding Assignment

Points possible: 75

URL to GitHub Repository: https://github.com/oalinares/Week3and4CodingAssignment

URL to Public Link of your Video:
https://www.youtube.com/watch?v=Bo1unJMuN7k&ab_channel=OscarLinares

---

Instructions:

1. Follow the **Coding Steps** below to complete this assignment.
    - 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.
    - Create a new repository on GitHub for this week's assignment and push your completed code to this dedicated repo.
    - Create a video showcasing your work:
        - In this video: record and present your project verbally while showing the results of the working project.
        - Easy way to Create a video: Start a meeting in Zoom, share your screen, open Eclipse with the code and your Console window, start recording & record yourself describing and running the program showing the results.
        - Your video should be a maximum of 5 minutes.
        - Upload your video with a public link.
        - Easy way to Create a Public Video Link: Upload your video recording to YouTube with a public link.
2. In addition, please include the following in your Coding Assignment Document:
    - The URL for this week's GitHub repository.
    - The URL of the public link of your video.
3. Save the Coding Assignment Document as a .pdf and do the following:
    - Push the .pdf to the GitHub repo for this week.
    - Upload the .pdf to the LMS in your Coding Assignment Submission.

Intro to Java Weeks 3-4 Coding Assignment

Coding Steps — Arrays and Methods

1. Create an array of int called ages that contains the following values: 3, 9, 23, 64, 2, 8, 28, 93.

    i. **int[] ages = {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.

    i. **int subtract = ages[ages.length-1] - ages[0];**

    ii. **System.out.println("This is the answer to subtract the value of the first element to the last: " + subtract);**

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

    i. **Simply add a new age to the end of 93, so we are going to go with 25. if we add 25 to the end of the array, we should get 22.**

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

    i. **int sum = 0;**

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

    iii. **sum += ages[i];}**

    iv. **int average = sum / ages.length;**

    v. **System.out.println("This is the answer to the average age in the array is: " + average);**

2. Create an array of String called names that contains the following values: “Sam”, “Tommy”, “Tim”, “Sally”, “Buck”, “Bob”.

    i. **String[] names = {"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.

    i. **int sumOfLetters = 0;**

    ii. **for (String letters : names) {**

    iii. **sumOfLetters += letters.length();}**

    iv. **int avgOfLetters = sumOfLetters / names.length;**

    v. **System.out.println("This is the average number of letters per name: " + avgOfLetters);**

    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.

i. **for (String firstName : names) {**

ii. **System.out.print(firstName.concat(" "));**

iii. **}**

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

   i. **To access the last element of the array, we are going to use the names array, so this array has 6 names, and so we can make the code dynamic we just use: names[names.length-1]**

   ii. **Even if the length changes, it will always return to the last element of the array.**

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

   i. **We are going to use the names example again for this, and the first element of an array will be names[0] due to zero being the representation of the initial element.**

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.

   i. **int[] nameLengths = new int[names.length];**

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

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

   iv. **System.out.println("These are the names and lengths: " + names[i] + " " + nameLengths[i] + " ");**

   v. **}**

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.

   i. **int sumOfNameLengths = 0;**

   ii. **for (int number : nameLengths) {**

   iii. **sumOfNameLengths += number;}**

   iv. **System.out.println("This is the sum of all the name lengths: " + sumOfNameLengths);**

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 expect the method to return “HelloHelloHello”).

   i. **public static String createRepeatOfWord(String word, int num) {**

   ii. **String result = "";**

   iii. **for (int i = 0; i < num; i++) {**

   iv. **result += word;**

```
v.                          }
vi.                         return result;
vii.                }
```

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).

```
i.  public static String createFullName(String firstName, String lastName) {
ii.                         String fullName = "";
iii.                        fullName = firstName + " " + lastName;
iv.                         return fullName;
v.                  }
```

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.

```
i.   public static boolean isSumOfArrayGreater(int[] array) {
ii.                         int sum = 0;
iii.                        for (int number : array) {
iv.                                 sum+= number;
v.                                  if (sum > 100) {
vi.                                         return true;
vii.                                        }
viii.
ix.                                 }
x.
xi.                                 return false;}
```

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

```
i.  public static double createAverageOfArray(double[] array2){
ii.                         double sumOfArray2 = 0;
iii.                for (double number : array2) {
iv.                         sumOfArray2+= number;
v.
vi.
```

```
vii.        }
viii.                   return sumOfArray2 / array2.length;    }
```

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.

```
i.  public static boolean isArray2GreaterThanArray3(double[] array2, double[] array3){
ii.                     if(createAverageOfArray(array2) >
    createAverageOfArray(array3)) {
iii.                        return true;
iv.
v.          }
vi.                 return false;
vii.        }
```

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.

```
i.  public static boolean willBuyDrink(boolean isHotOutside, double moneyInPocket) {
ii.                     if(isHotOutside == true && moneyInPocket > 10.50) {
iii.                        return true;
iv.
v.                      }
vi.                     return false;
vii.        }
```

13. Create a method of your own that solves a problem. In comments, write what the method does and why you created it. Write a method that converts miles into kilometers, and gives you the average kilometers of the array**.**

```
i.  public static double milesToAverageKilometers(double[] milesBikedInMi) {
ii.                     double sum = 0;
iii.                    double sumKm = 0;
iv.                     for(double miles : milesBikedInMi) {
v.                          sum += miles;
vi.                         sumKm = sum * 1.609344;
vii.
```

```
viii.
ix.                                    }
x.                                     return sumKm / milesBikedInMi.length;
xi.    }
```