

```

package week3andfour;

import java.lang.reflect.Array;
import java.util.Arrays;
import java.util.OptionalDouble;

public class weekThree {

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

//1.
        int[] ages = {3, 9, 23, 64, 2, 8, 28, 93};
        //a. and b. subtracts 1st and last element and b) added a new int
        (36)

        { System.out.println(ages[0] - (ages[ages.length - 1])); }
        int sum = 0;
        int[] newAges = {3, 9, 23, 64, 2, 8, 28, 93, 36};
        { System.out.println(newAges[0] - (newAges [newAges.length
- 1])); }
        }
        //c. average age
        for (int age : ages)
            sum += age;
        {System.out.println(sum / ages.length);
    }

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

            int sumLetters = 0;
            for (int i = 0; i < names.length; i++) {
                sumLetters += names[i].length();
                System.out.println(sumLetters);
            }
            sum = sumLetters / names.length;
            {System.out.println(sum);}

        // 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.

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

```

```

        concatNames += names[i] + " "; }
        System.out.println(concatNames);

//3. How do you access the last element of any array?
        { System.out.println("lastElement =
array[array.length(-1)");
        }
//4. How do you access the first element of any array?
        { System.out.println("firstElement =
array[0]");
        }

//5. 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[names.length];
        for (int i = 0; i < names.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.
        sum = 0;
        for (int i = 0; i < nameLengths.length; i++)
            sum += nameLengths[i];

        { System.out.println(sum);
        }

//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").

        System.out.println(repeatWord("Hello", 5));

//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).
        String firstName = "Dolly";
        String lastName = "Parton";
        String fullName = theFullName(firstName, lastName);
        System.out.println(fullName);

//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 100
    System.out.println(sumGreater(ages));
```

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

```
    double[] doubleArr = {3.4, 9.0, 3.7, 2.9, 100.8};
```

```
    System.out.println(getAvgDoubleArr(doubleArr));
```

```
//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.
```

```
    double[] blueDoubleArr = {3.4, 2.7, 80.4};
```

```
    double[] redDoubleArr = {1.4, 7.3, 89.3};
```

```
    System.out.println(isBlueArrAvgGreater(blueDoubleArr,
redDoubleArr));
```

```
//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.
```

```
    boolean isHotOutside = true;
```

```
    double moneyInPocket = 11.50;
```

```
    System.out.println(willBuyDrink(isHotOutside,
moneyInPocket));
```

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

```
    double isUnderLegalLimit = 0.8;
```

```
    boolean hasLicense = true;
```

```
    System.out.println(canDrive(isUnderLegalLimit,
hasLicense));}
```

```
    // Checks to see what a persons alcohol level is and
if they have license to determine if they can drive or not
```

```
    //7.
```

```
        public static String repeatWord(String word, int n) {
            String repeatWord = " ";
            for (int i = 0; i < n; i++) {
                repeatWord += word;
            }
            return repeatWord;
        }
```

```
//8.
```

```
        public static String theFullName(String firstName, String
lastName) {
```

```

        return firstName + " " + lastName;
    }

//9.
    public static Boolean sumGreater(int[] arr) {
        return Arrays.stream(arr).sum() > 100;
    }
//10.
    public static Double getAvgDoubleArr(double[] arr) {
        double getAvgDoubleArr = 0;
        for (double i = 0; i < Array.getLength(arr); i++) {
            return getAvgDoubleArr;
        }
    }
//11.
    public static boolean isBlueArrAvgGreater(double[]
blueArr, double[] redArr) {
        return getAvgDoubleArr(blueArr) >
getAvgDoubleArr(redArr);
    }
//12.
    public static boolean willBuyDrink(boolean isHotOutside,
double moneyInPocket) {
        if (isHotOutside == true && moneyInPocket > 10.50)
            return true;
        else return false;
    }
//13.
    public static boolean canDrive(double isUnderLegalLimit,
boolean hasLicense) {
        if (isUnderLegalLimit <= 0.8 && hasLicense == true)
            return true;
        else return false;
    }
}

```