PROMINEO TECH

Intro to Java Weeks 3-4 Coding Assignment

Points possible: 75

URL to GitHub Repository: https://github.com/BNHcoder/HWWeek-3-4-

URL to Public Link of your Video:
https://studio.youtube.com/channel/UCR5qmkd4v1Wx8qUKizDRyKA/videos/upload?filter=%5B%5D&sort=%7B%22columnType%22%3A%22date%22%2C%22sortOrder%22%3A%22DESCENDING%22%7D

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.

PROMINEO TECH

Intro to Java Weeks 3-4 Coding Assignment

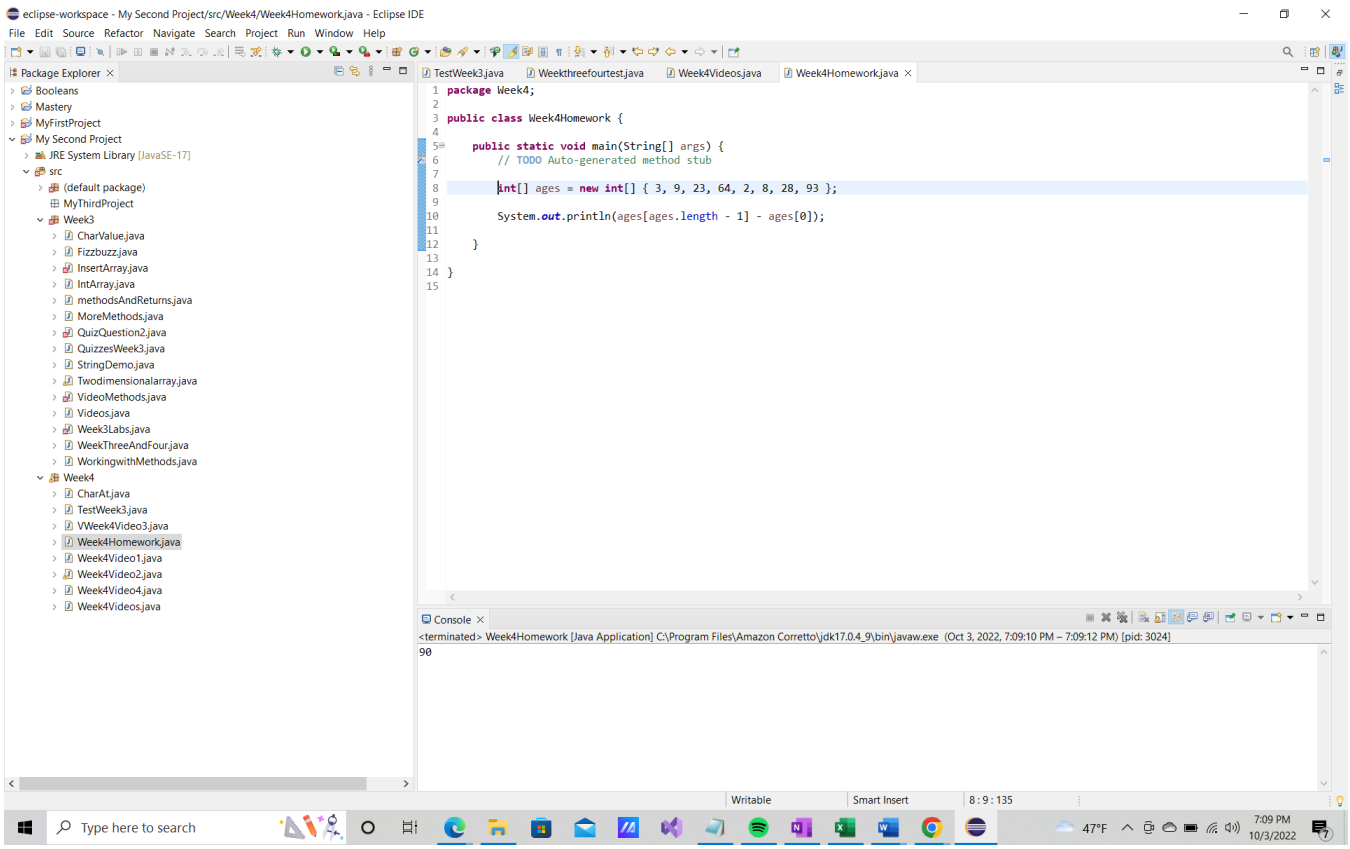
Intro to Java Weeks 3-4 Coding Assignment

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.

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

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

//System.out.println(ages[i]);
```

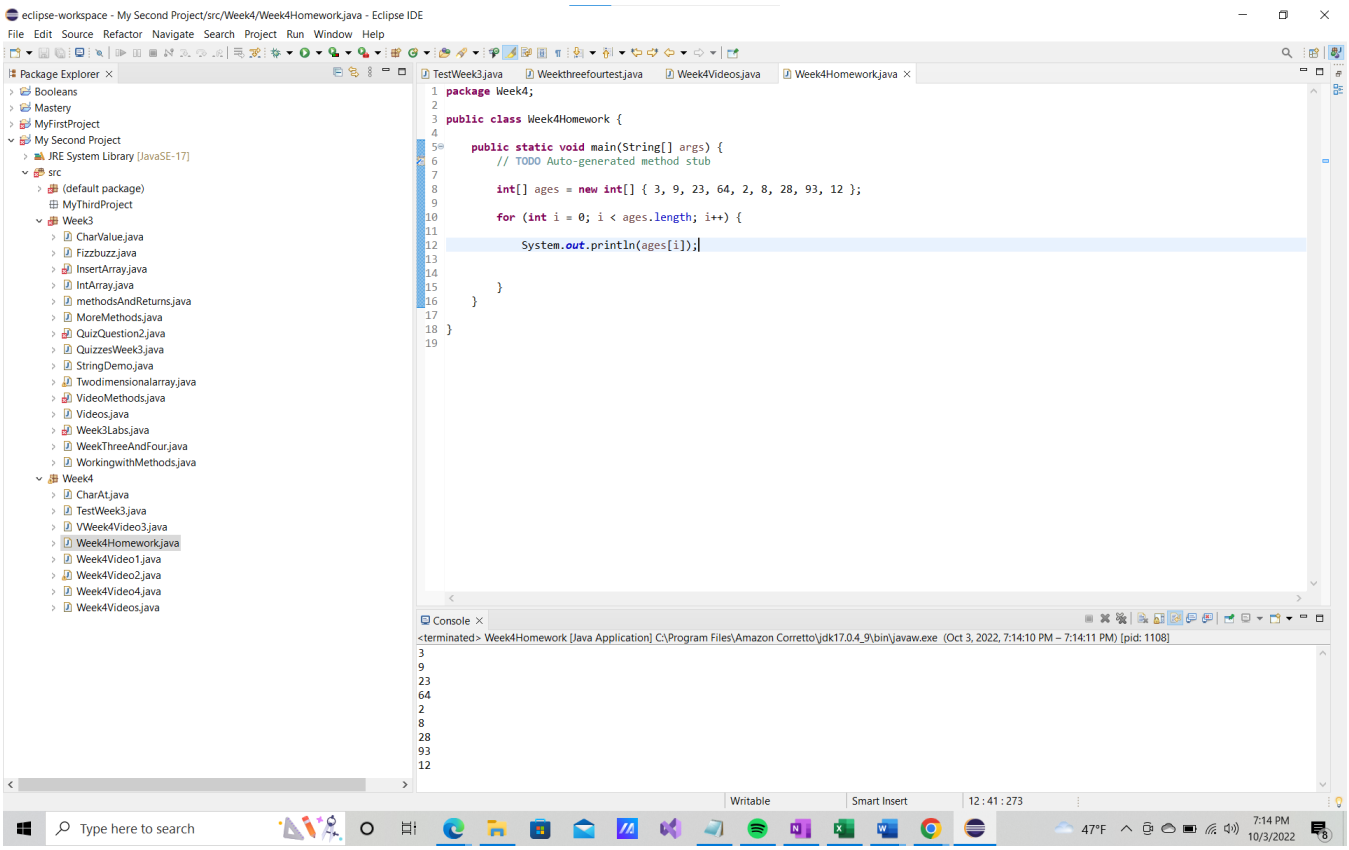
   a. Programmatically subtract the value of the first element in the array from the value in the last element of the array (i.e. do not use ages[7] in your code). Print the result to the console.




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

System.out.println(ages[ages.length-1] - ages[0]);

Console Prints 90      result of 93 - 90
```

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

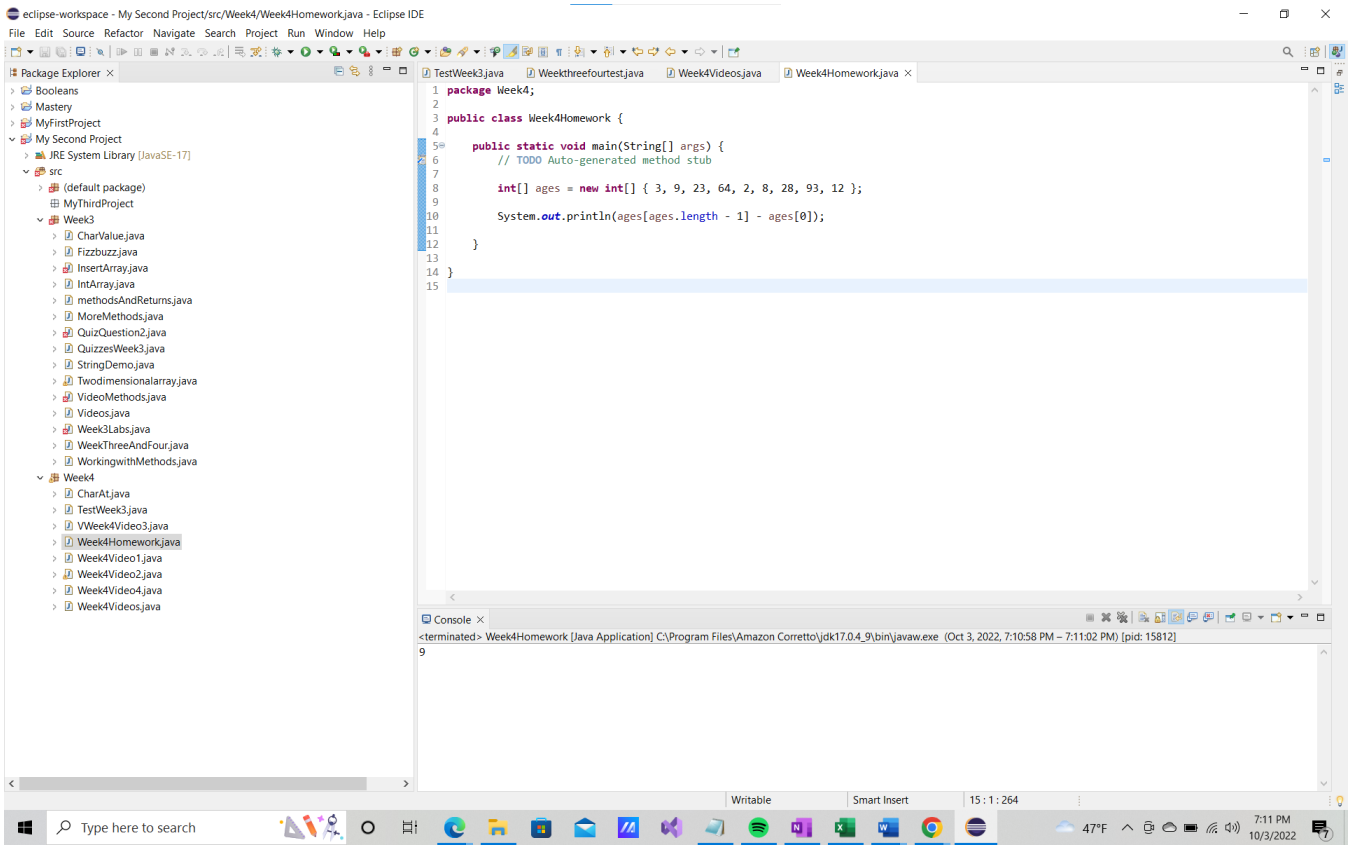
i.




```
int[] ages = new int[]{3,9,23,64,2,8,28,93,12};

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

System.out.println(ages[i]);


Result Console
3
9
23
64
2
8
28
```

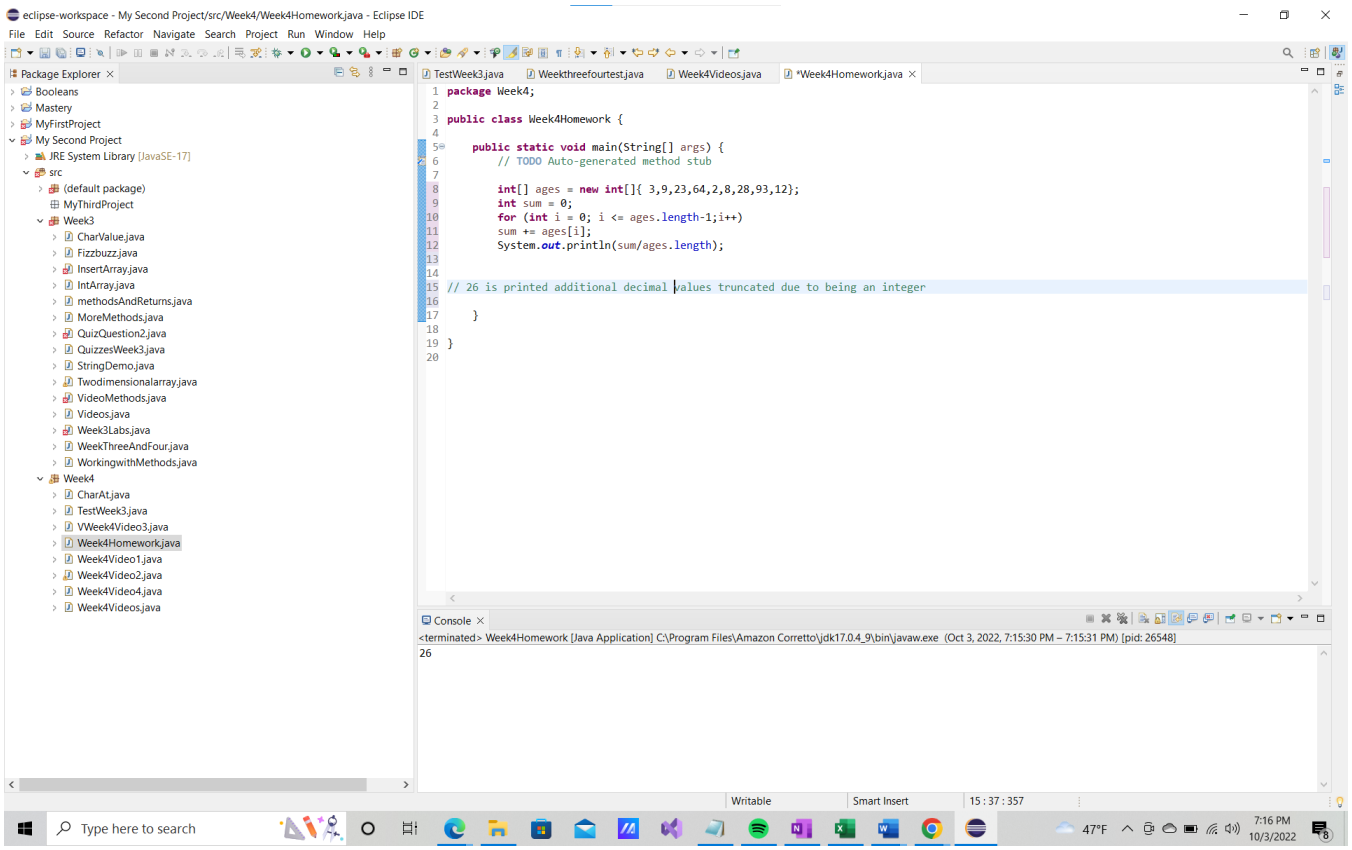
```
93
12
```

ii.




```
int[] ages = new int[]{3,9,23,64,2,8,28,93,12};

System.out.println(ages[ages.length-1] - ages[0]);

Result Console

9
```

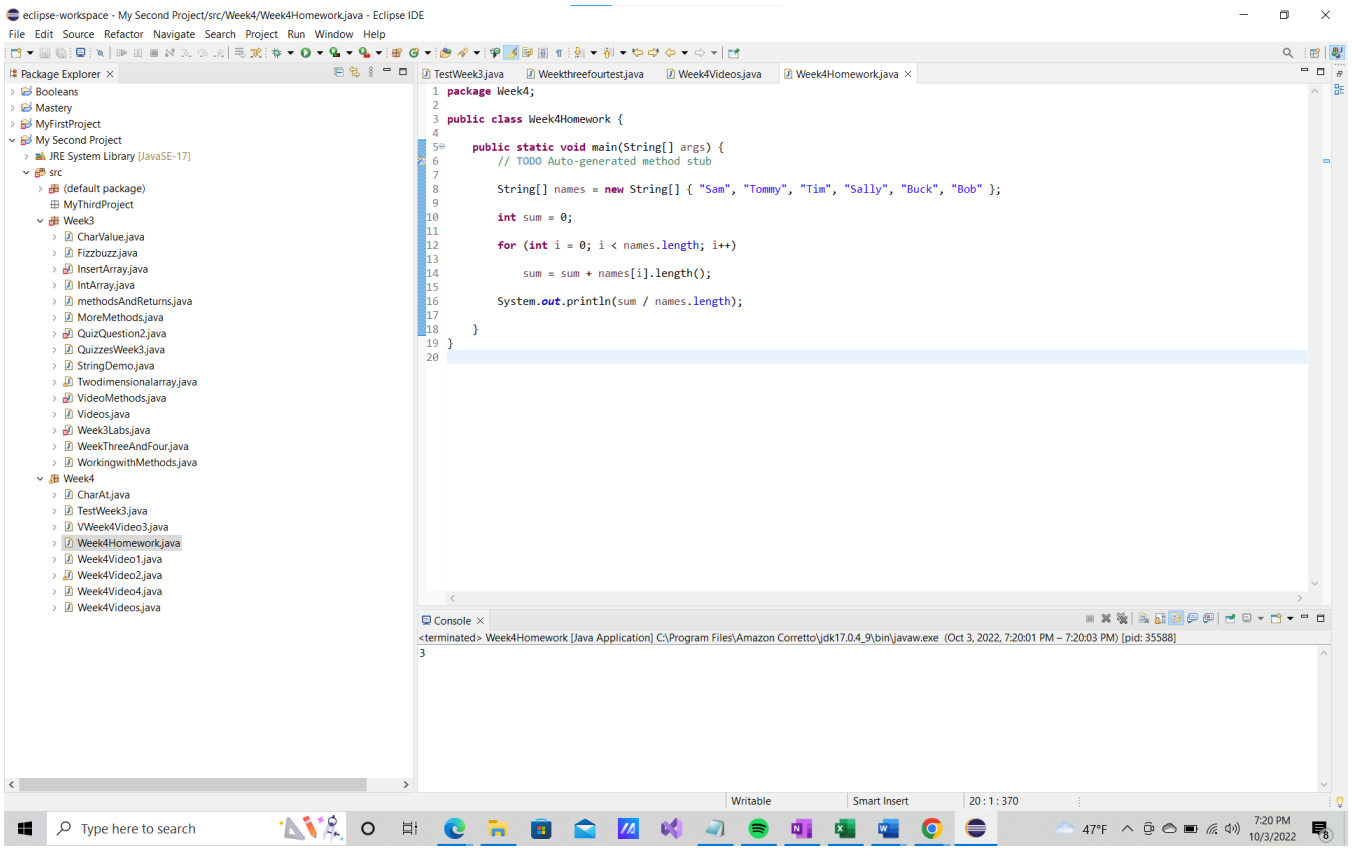
c. Use a loop to iterate through the array and calculate the average age. Print the result to the console.




```
int[] ages = new int[]{ 3,9,23,64,2,8,28,93,12};

int sum = 0;

for (int i = 0; i <= ages.length-1;i++)

sum += ages[i];

System.out.println(sum/ages.length);

      }

            }
result Console 26 //value is set at integer, decimals will not print
```

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.




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

int sum = 0;

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

sum = sum + names[i].length();

System.out.println(sum /names.length);
```

```
result Console 3 //value is set at integer, decimals will not print
```

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

String concatName= names[0];
```

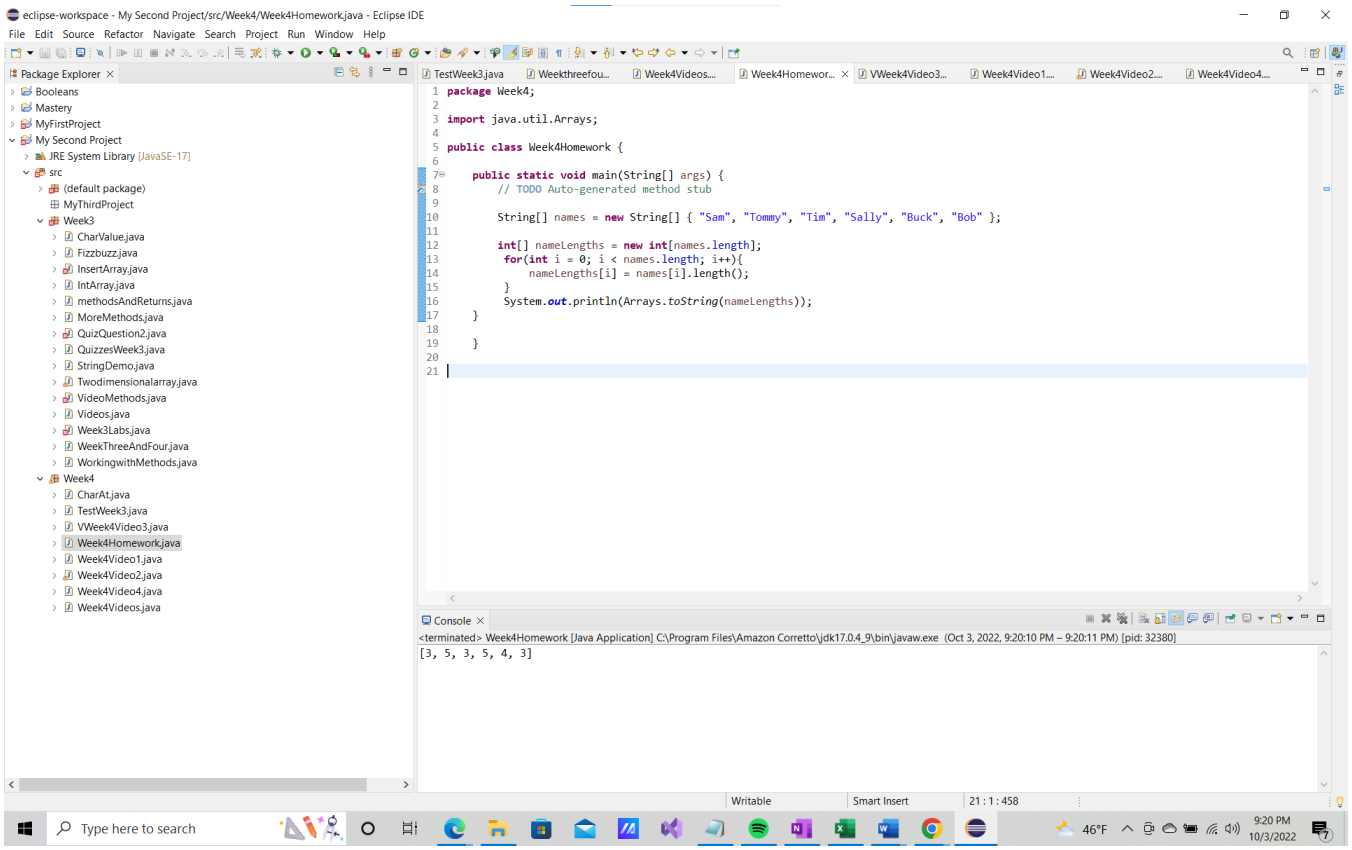
```
for(int i = 1; i <=names.length - 1; i++)

concatName = concatName.concat(" " + names[i]);

System.out.println(concatName);
```

```
result Console Sam Tommy Tim Sally Buck Bob
```

3. How do you access the last element of any array?
   lastElement = array[array.length-1];

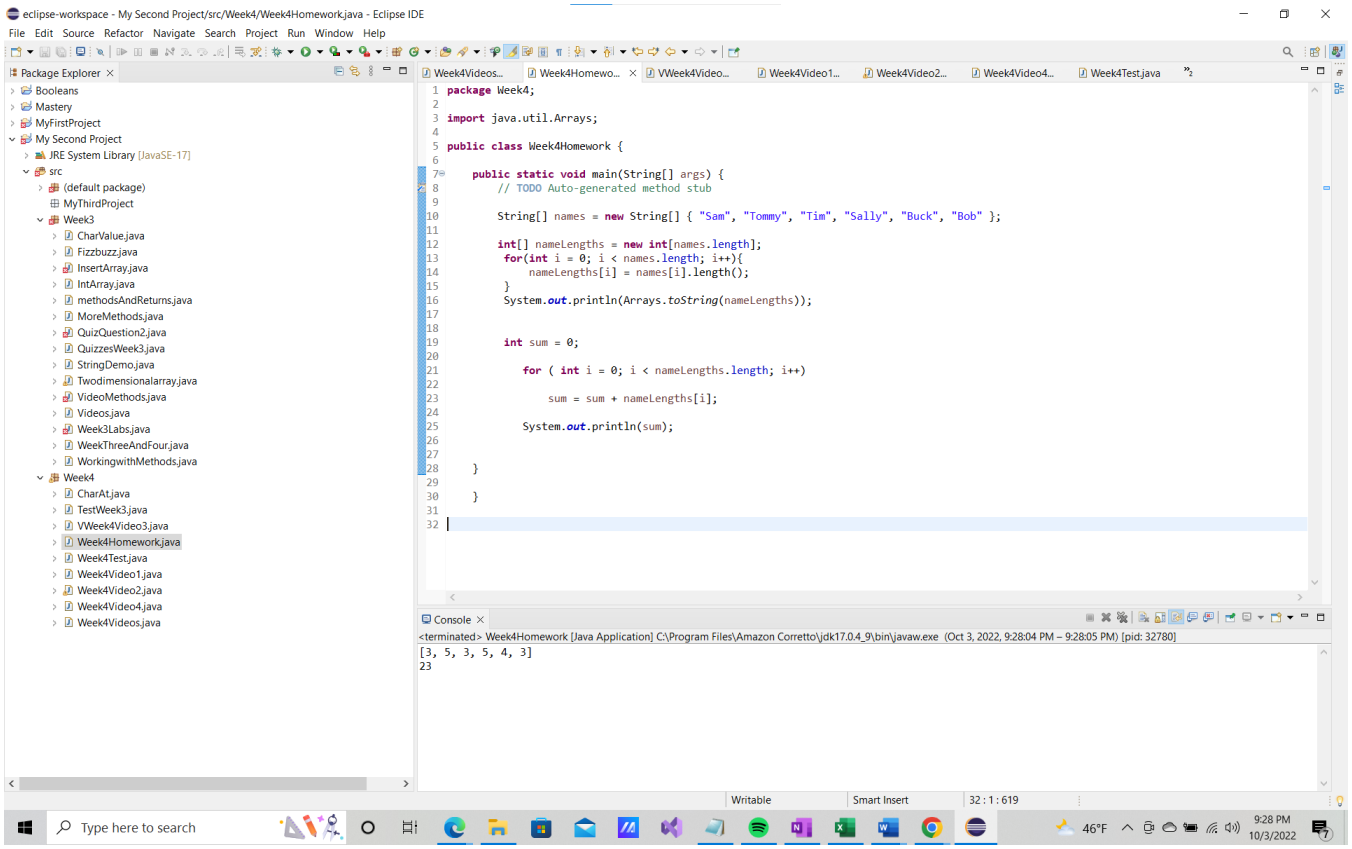
4. How do you access the first element of any array?
   firstElement = numbers[0];

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.

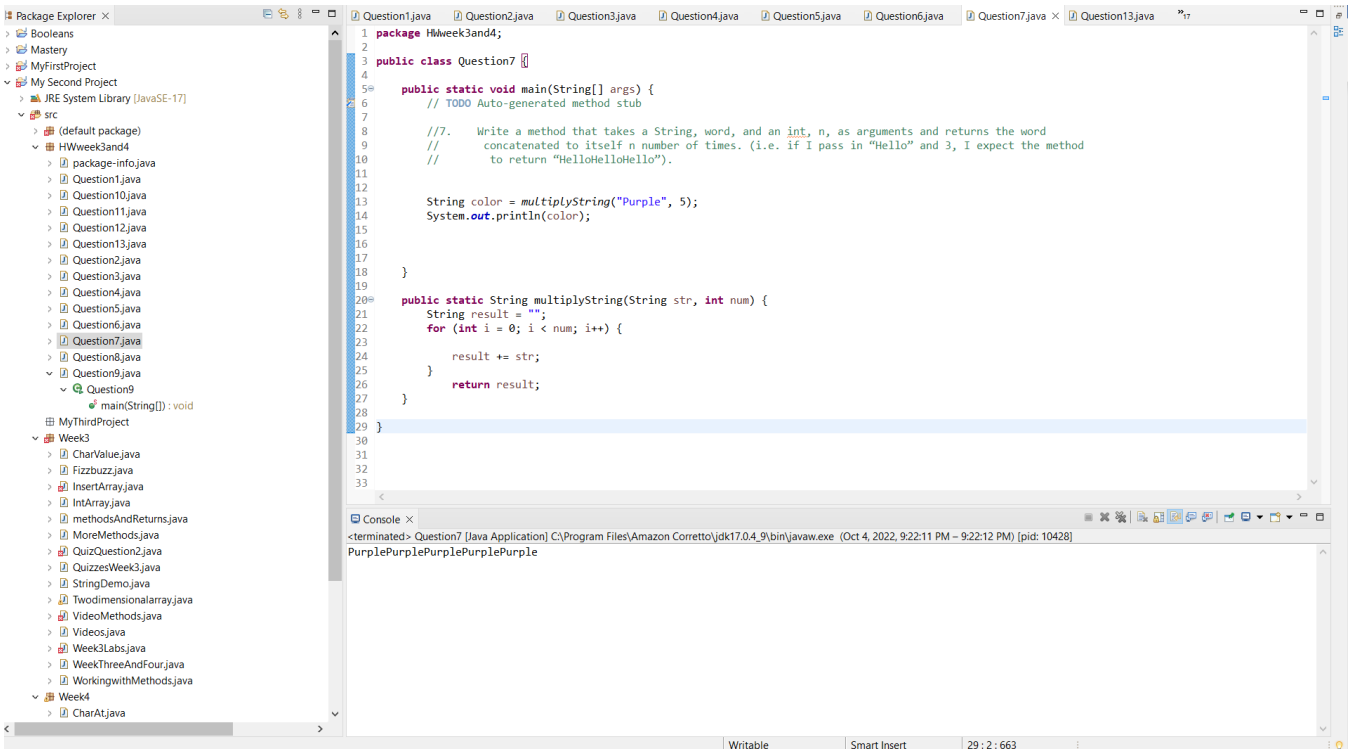



```
String[] names = new String[] { "Sam", "Tommy", "Tim", "Sally", "Buck", "Bob" };

            int[] nameLengths = new int[names.length];
             for(int i = 0; i < names.length; i++){
                 nameLengths[i] = names[i].length();
             }
             System.out.println(Arrays.toString(nameLengths));
    }

    }
```

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 expect the method to return “HelloHelloHello”).

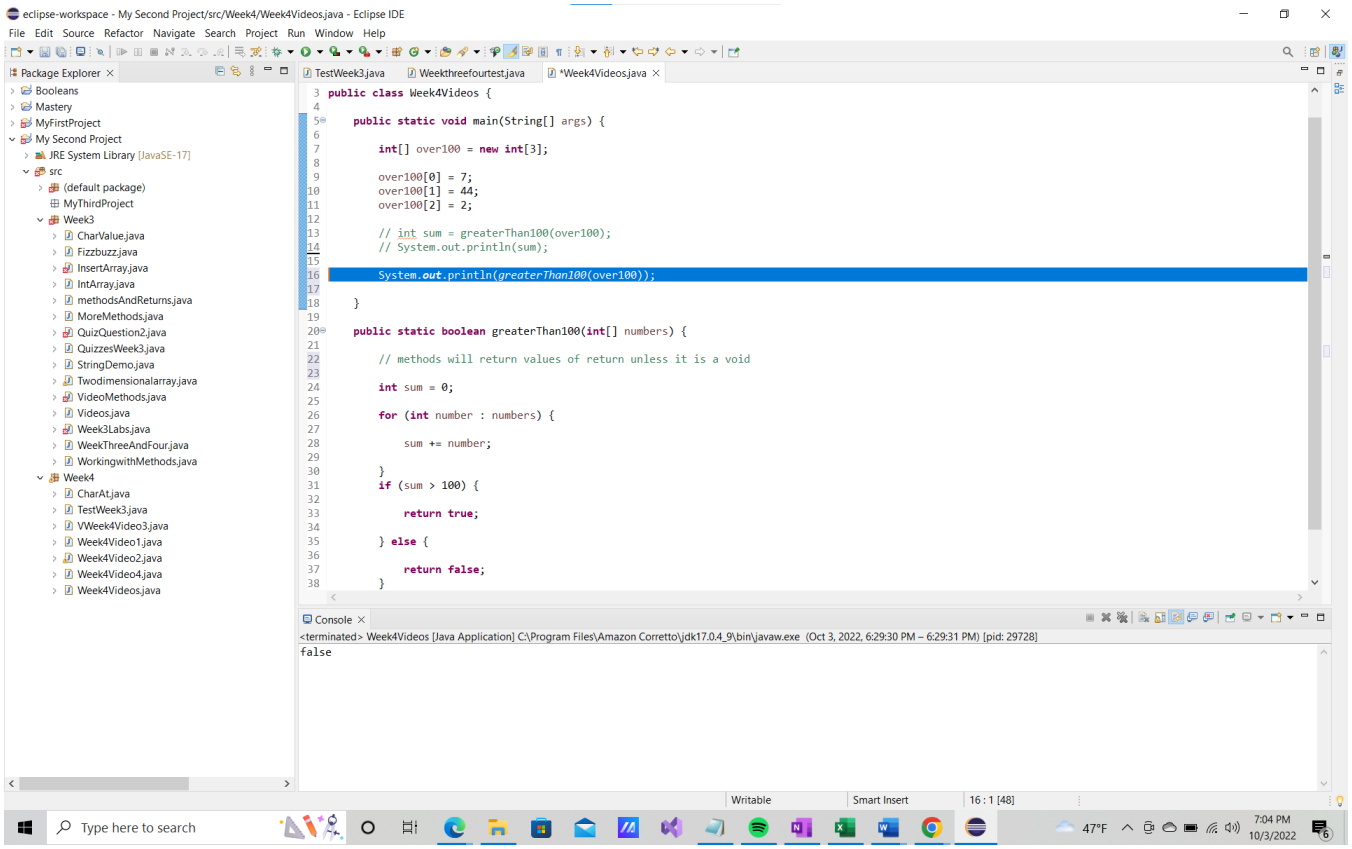



```
String color = multiplyString("Purple", 5);
System.out.println(color);



// method is below Note method is to be written outside the main
public static String multiplyString(String str, int num) {
String result = "";
for (int i = 0; i < num; i++) {

    result += str;
}
        return result;
```

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 = "Brittany";
        String lastName = "Hilton";

System.out.println(firstName.concat(" " + lastName));
```

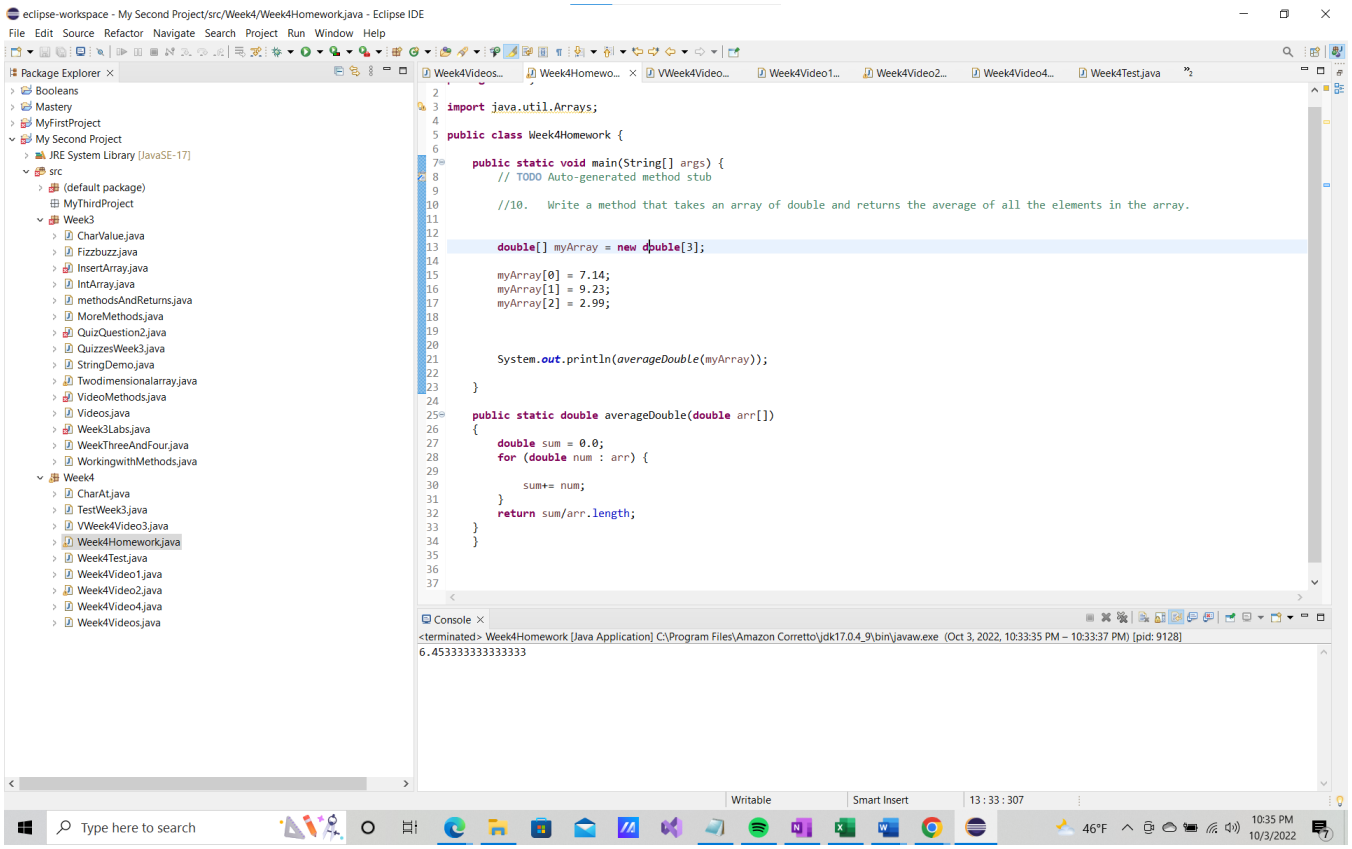
9. Write a method that takes an array of int and returns true if the sum of all the ints in the array is greater than 100.




```
        int[] over100 = new int[3];

        over100[0] = 7;
        over100[1] = 44;
        over100[2] = 2;


        System.out.println(greaterThan100(over100));

    }

    public static boolean greaterThan100(int[] numbers) {

        // methods will return values of return unless it is a void

        int sum = 0;
```

```
        for (int number : numbers) {

                sum += number;

        }
        if (sum > 100) {

                return true;

        } else {

                return false;
```

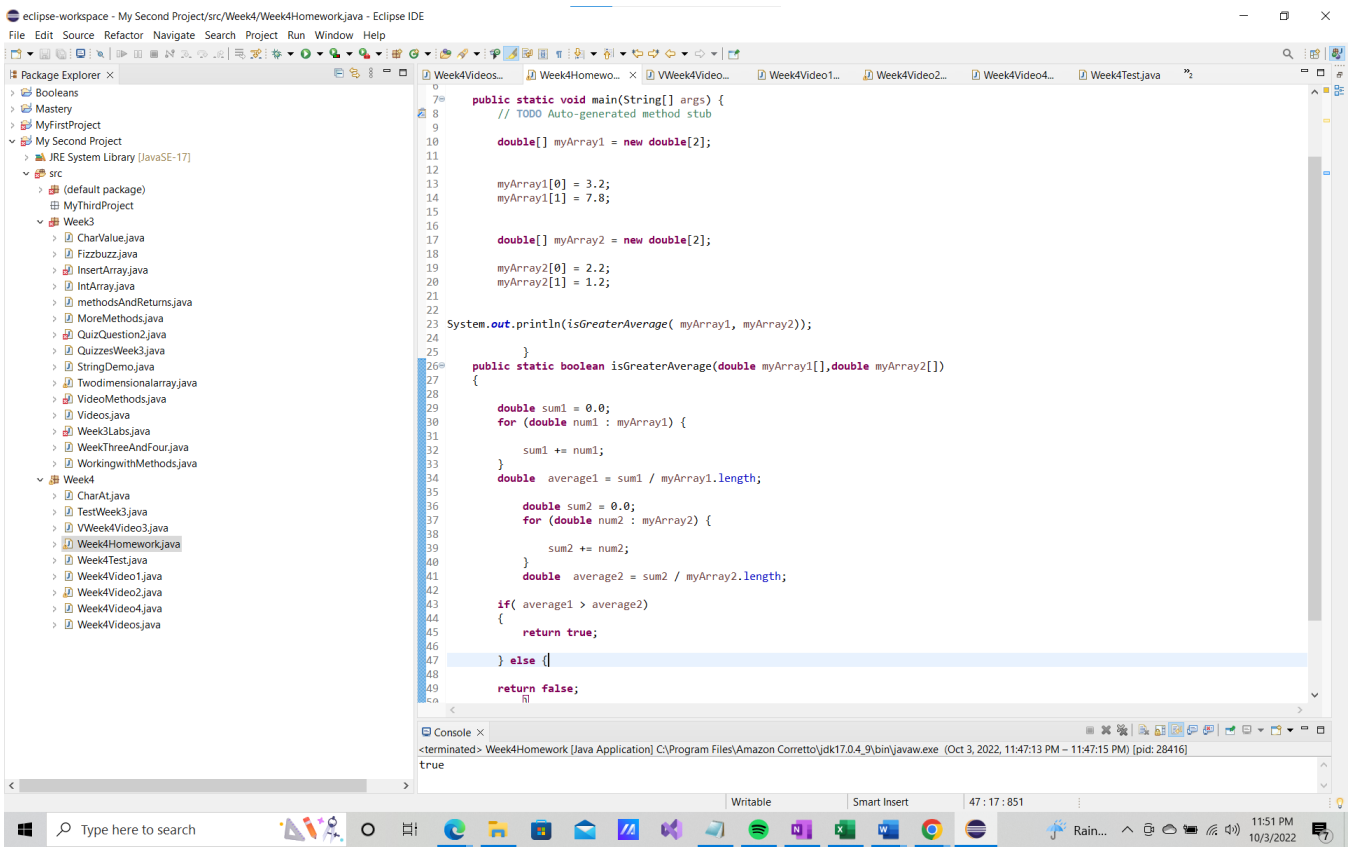
10. Write a method that takes an array of double and returns the average of all the elements in the array.




```
double[] myArray = new double[3];

        myArray[0] = 7.14;
        myArray[1] = 9.23;
        myArray[2] = 2.99;
```

```
        System.out.println(averageDouble(myArray));

    }

    public static double averageDouble(double arr[])
    {
            double sum = 0.0;
            for (double num : arr) {

                    sum+= num;
            }
                    return sum/arr.length;
```

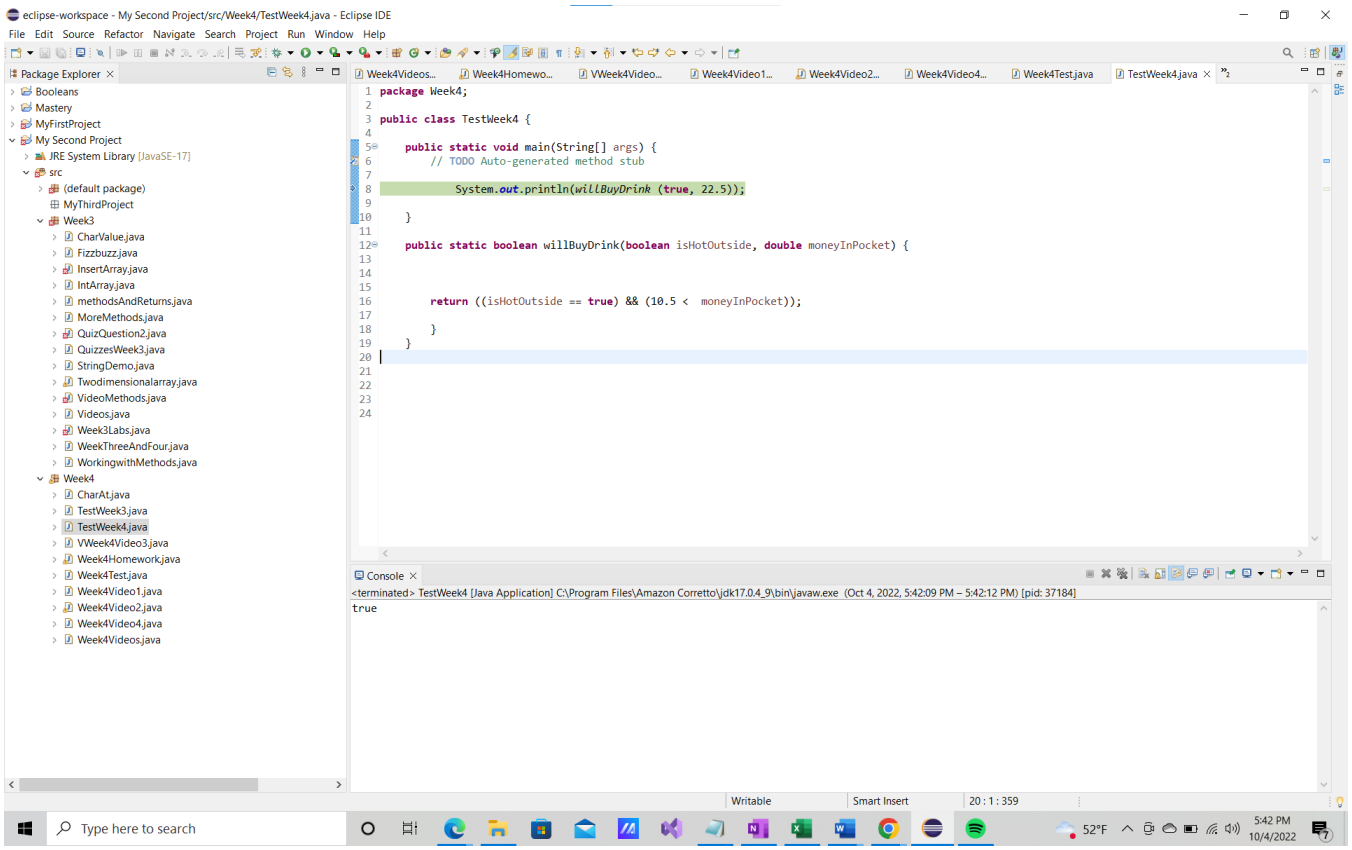
11. Write a method that takes two arrays of double and returns true if the average of the elements in the first array is greater than the average of the elements in the second array.




```
double[] myArray1 = new double[2];
```

```
		myArray1[0] = 3.2;
		myArray1[1] = 7.8;


		double[] myArray2 = new double[2];

		myArray2[0] = 2.2;
		myArray2[1] = 1.2;


System.out.println(isGreaterAverage( myArray1, myArray2));

			}
	public static boolean isGreaterAverage(double myArray1[],double myArray2[])
	{

		double sum1 = 0.0;
		for (double num1 : myArray1) {

			sum1 += num1;
		}
		double average1 = sum1 / myArray1.length;

			double sum2 = 0.0;
			for (double num2 : myArray2) {

				sum2 += num2;
			}
			double		average2 = sum2 / myArray2.length;

		if( average1 > average2)
		{
			return true;

		} else {

		return false;
			}

	}
		}
```

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.



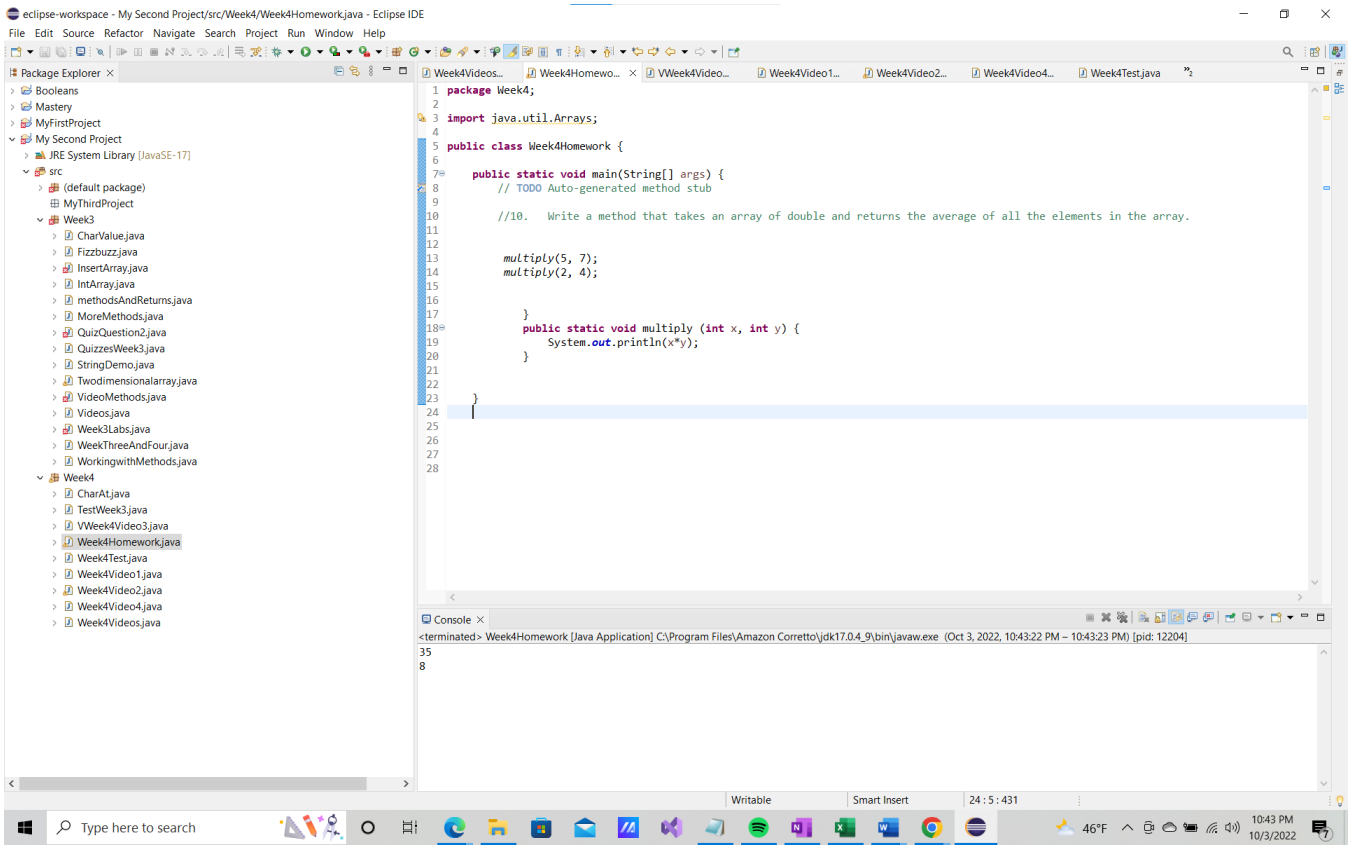

```
System.out.println(willBuyDrink (true, 22.5));

    }

    public static boolean willBuyDrink(boolean isHotOutside, double moneyInPocket) {


        return ((isHotOutside == true) && (10.5 <  moneyInPocket));

        }
        }


    Console true
```

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

PROMINEO TECH

Intro to Java Weeks 3-4 Coding Assignment

Create a method that will multiply TWO integers and give a result. The method multiplies two integers together. It was created as a way to sum multiple quantities.




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

 multiply(5, 7);
 multiply(2, 4);


	}
	public static void multiply (int x, int y) {
		System.out.println(x*y);
	}

}
```

Console print

35

8