Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-450-M2024/it114-module-2-java-problems/grade/jah89

IT114-450-M2024 - [IT114] Module 2 Java Problems

Submissions:

Submission Selection

1 Submission [active] 6/1/2024 11:45:29 AM

Instructions

COLLAPSE

Overview Video: https://youtu.be/4M8Di5jrcZQ

Guide:

1. Make sure you're in the main branch locally and `git pull origin main` any pending changes.
2. Make a new branch per the recommended branch name below (`git checkout -b ...`).
3. Grab the template code from https://gist.github.com/MattToegel/fdd2b37fa79a06ace9dd259ac82728b6.
4. Create individual Java files for each problem and save the files inside a subfolder of your choice.
    1. They should end with the file extension in lowercase `.java`.
5. Move the unedited template files to GitHub.
    1. `git add .`
    2. `git commit -m "adding template files"`
    3. `git push origin branch_name` (see below).
    4. Create and open a pull request from the homework branch to main (leave it open until later steps).
6. Note: As you work, it's recommended to add/commit at least after each solution is done (i.e., 3+ times in this case).
    1. Make sure the files are saved before doing this.
7. Fill in the items in the worksheet below (save as often as necessary).
8. Once finished, export the worksheet.
9. Add the output file to any location of your choice in your repository folder (i.e., a `Module2` folder).
10. Check that git sees it via `git status`.
11. If everything is good, continue to submit.

1. Track the file(s) via `git add`.
2. Commit the changes via `git commit` (don't forget the commit message).
3. Push the changes to GitHub via `git push` (don't forget to refer to the proper branch).
4. Create a pull request from the homework related branch to main (i.e., main <- "homework branch").
5. Open and complete the merge of the pull request (it should turn purple).
6. Locally checkout main and pull the latest changes (to prepare for future work).

12. Take the same output file and upload it to Canvas.

Branch name: M2-Java-Problems

Tasks: 6 Points: 10.00

Problem 1 (3 pts.)

^COLLAPSE^

^COLLAPSE^

Task #1 - Points: 1

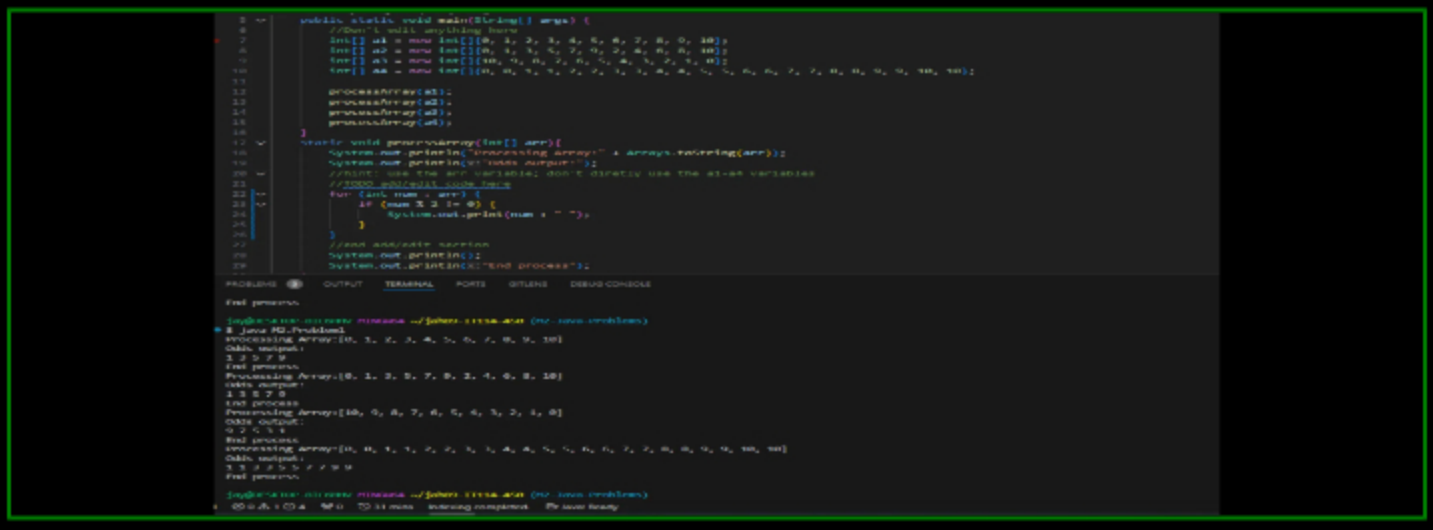
Text: Screenshot of the Problem 1 Solved Code and Output

Details:

Only make edits where the template code mentions.

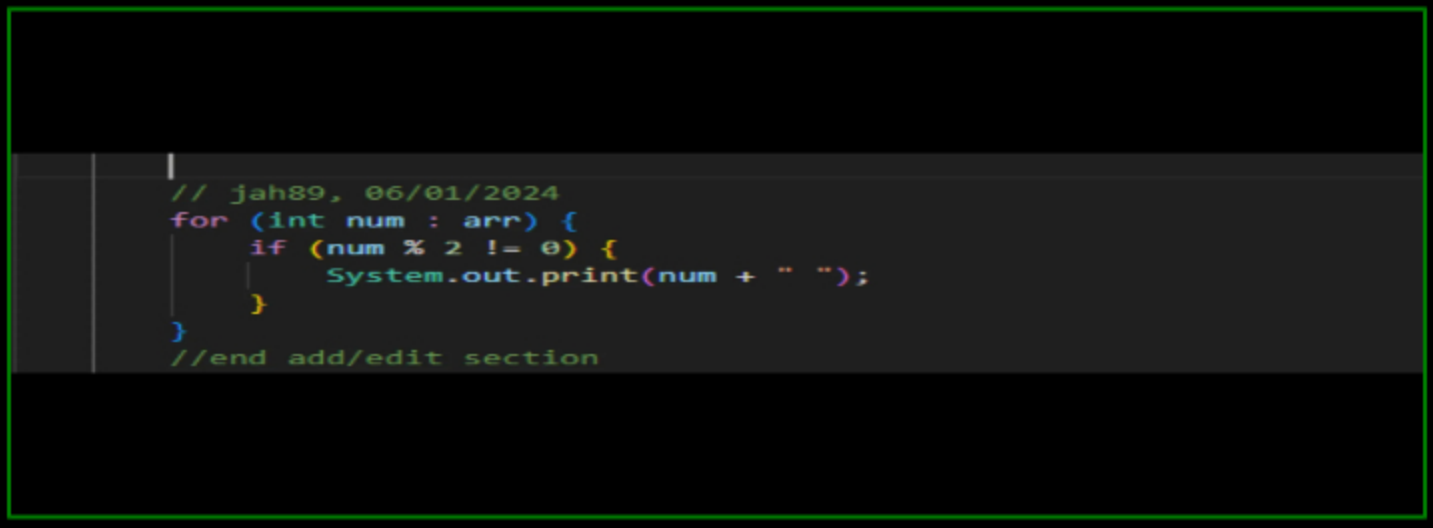
Solution should ensure that any passed in array will have only the odd values output.
Requires at least 2 screenshots (code + output from terminal)

#1) Screenshot the output of the solved problem



Caption (required) ✓
Describe/highlight what's being shown
Showing

#2) Screenshot the code solution (ucid/date must be included as a comment)



```
// jah89, 06/01/2024
for (int num : arr) {
    if (num % 2 != 0) {
        System.out.print(num + " ");
    }
}
//end add/edit section
```

Caption (required) ✓
Describe/highlight what's being shown
Showing code added to problem1

Explanation (required) ✓
Explain in concise steps how this logically works
EDIT RESPONSE
I used for-each loop to iterate over the "arr" array. In this case the num variable is just the current element in the array during that iteration. Inside the if statement is checking for odd numbers in the array. The num%2 shows the remainder when the "num" is divided by 2. So because of this if it does not equal zero it is a odd number. So then if it's an odd number it goes to print statement and then prints the number("num") from array that gave a remainder and adds a space after for formatting.

Problem 2 (3 pts.)
COLLAPSE

COLLAPSE

Task #1 - Points: 1
Text: Screenshot of the Problem 2 Solved Code and Output

Details:
Only make edits where the template code mentions.

Solution should ensure that any passed in array will have its values summed AND the final result converted to two decimal places (i.e. 0.10, 1.00, 1.01)

converted to two decimal places (i.e., 0.10, 1.00, 1.01).
Requires at least 2 screenshots (code + output from terminal)

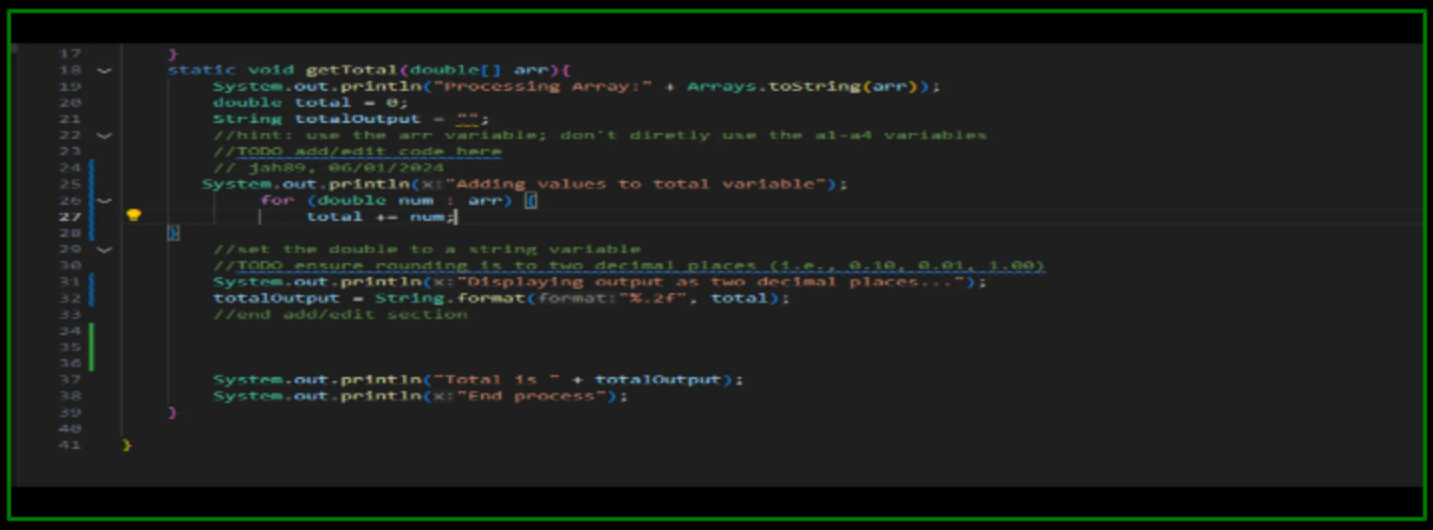
#1) Screenshot the output of the solved problem

Caption (required) ✓

Describe/highlight what's being shown

SHowing output for Problem2

#2) Screenshot the code solution (ucid/date must be included as a comment)



```
    }
    static void getTotal(double[] arr){
        System.out.println("Processing Array:" + Arrays.toString(arr));
        double total = 0;
        String totalOutput = "";
        //hint: use the arr variable; don't diretly use the a1-a4 variables
        //TODO add/edit code here
        // jah89, 06/01/2024
        System.out.println(x:"Adding values to total variable");
            for (double num : arr) {
                total += num;
            }
        //set the double to a string variable
        //TODO ensure rounding is to two decimal places (i.e., 0.10, 0.01, 1.00)
        System.out.println(x:"Displaying output as two decimal places...");
        totalOutput = String.format(format:"%.2f", total);
        //end add/edit section


        System.out.println("Total is " + totalOutput);
        System.out.println(x:"End process");
    }

}
```

Caption (required) ✓

Describe/highlight what's being shown

Code for Problem2

Explanation (required) ✓

Explain in concise steps how this logically works

EDIT RESPONSE

For the first part of the problem I wrote the message "Adding values to total variable" to make the readability better in the terminal when I run the program. I then used a for each loop to iterate through the array. I then used total += num to add the current element of the array to the variable total. This will add every element of the array and just put it into the variable total giving me the sums of the values. For the second part I wrote a print message to show displaying output as two decimal places. I then took the variable totalOutput and used String.format to show the double rounded to two decimal places. I used %.2f because the % is the start of the formatting and the .2 represents the two decimal places and then the f to represent a floating number. Then it is assigned to the totalOutput variable.

Problem 3 (3 pts.)

^COLLAPSE^

^COLLAPSE^

Task #1 - Points: 1

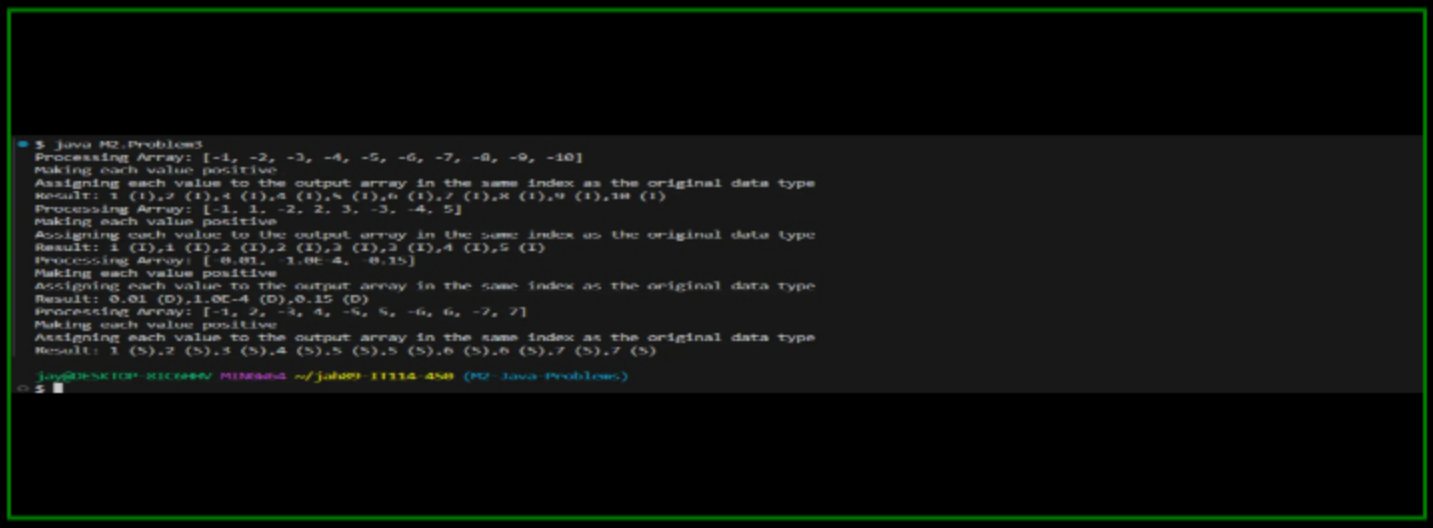
Text: Screenshot of the Problem 2 Solved Code and Output

Details:

Only make edits where the template code mentions.

Solution should ensure that any passed in array will have its values converted to a positive version of the value AND converted back to the original data type.
Requires at least 2 screenshots (code + output from terminal)

#1) Screenshot the output of the solved problem



Caption (required) ✓

Describe/highlight what's being shown

The output for problem 3 making each value positive

#2) Screenshot the code solution (ucid/date must be included as a comment)

Caption (required) ✓

Describe/highlight what's being shown

Showing code solution to problem3

Explanation (required) ✓

Explain in concise steps how this logically works

PREVIEW RESPONSE

I created a for loop to iterate through each element of the array which is called "arr". I then wanted to check the data type so I made the variable "value" to store the current element from array and used InstanceOf to check the data type of the element. I made sure if it was a double or an integer to make sure to take its absolute value by using Math.abs(). I then store these values in the output array. If the element is a string it checks for a "-" which would show the element is negative and then use strValue.substring(1) which is first index which would remove the negative sign from the string. All these values are stored in the output array at the same index.

Reflection (1 pt.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Reflect on your experience

Details:

Talk about any issues you had, how you resolved them, and anything you learned during this process.

Provide concrete details/examples. At least a few sentences.

Response:

I had a little trouble with problem3 I wasn't too familiar with handling data types properly, but watched a couple of youtube tutorials and got the hang of it and learned more about using "instanceof" to check data types. Besides those issues for problem 1 and 2 I had no issues and was able to refresh on my java skills and prepare myself for harder

issues for problem 1 and 2 I had no issues and was able to refresh on my java skills and prepare myself for harder tasks coming up.

COLLAPSE

Task #2 - Points: 1

Text: Include the pull request link for this branch

Details:

The correct link will end with /pull/ and a number.

URL #1

https://github.com/jah89/jah89-IT114-450/pull/4

COLLAPSE

Task #3 - Points: 1

Text: Add Screenshot of Wakatime

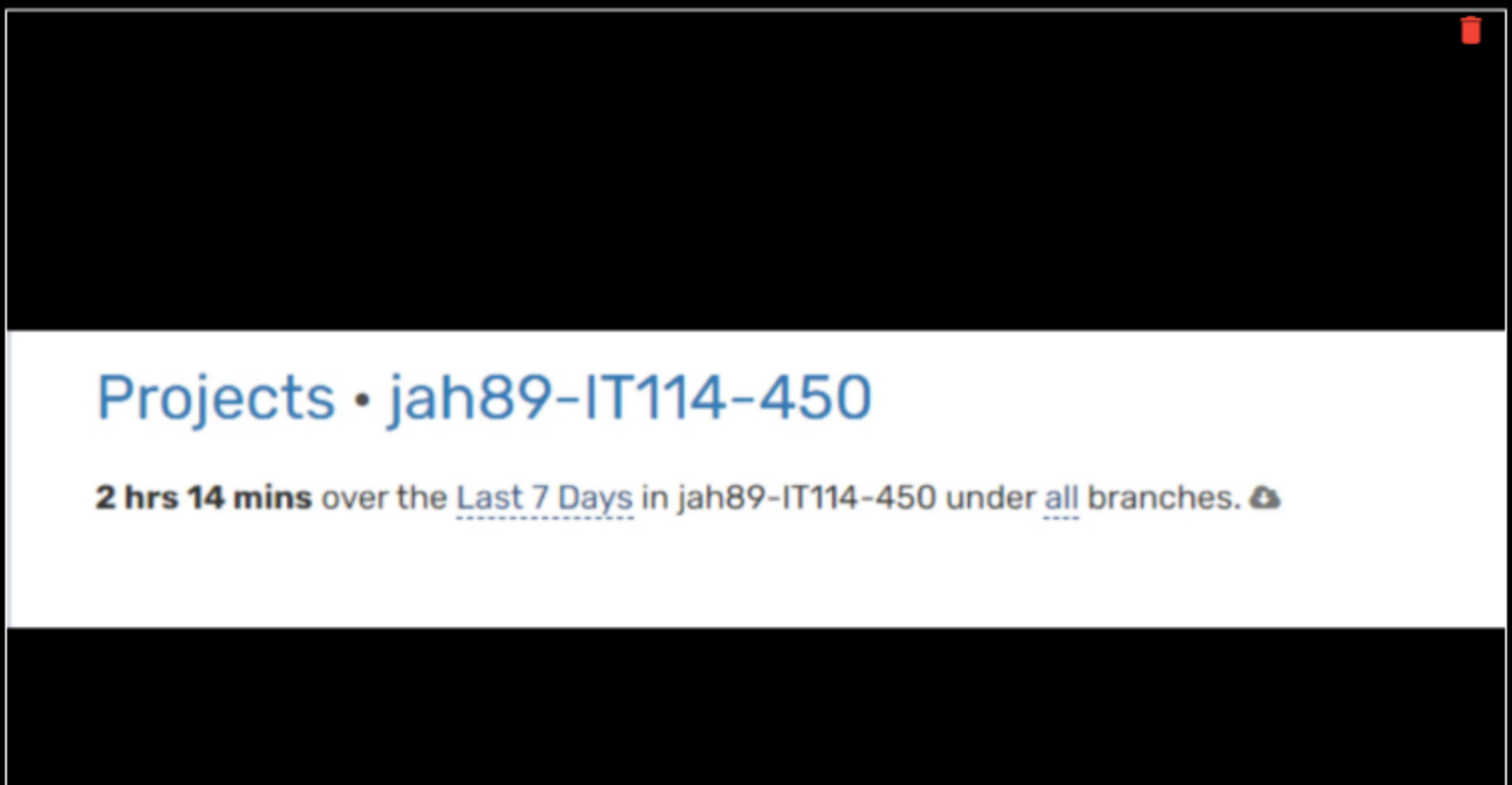
Details:

Note: The duration of time isn't directly related to the grade, the goal is to just make sure time is being tracked

Task Screenshots:

Gallery Style: Large View

Small Medium Large



How much time spent over past 7 days

Files

1 hr 16 mins	M2/Problem3.java
29 mins	M2/Problem2.java
25 mins	M2/Problem1.java
3 mins	.gitignore
2 secs	...started_IT114-450-M2024.pdf

How much time spent on each file

End of Assignment