Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-005-F2024/it114-module-2-java-problems/grade/kh465

Course: IT114-005-F2024
Assigment: [IT114] Module 2 Java Problems
Student: Keven H. (kh465)

Submissions:

Submission Selection

1 Submission [submitted] 9/22/2024 2:30:19 PM

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. Create a folder in your local repo called `Module2`
4. Grab the template code from https://gist.github.com/MattToegel/fdd2b37fa79a06ace9dd259ac82728b6.
5. Create individual Java files for each problem and save the files inside the `Module2` folder.
    1. They should end with the file extension in lowercase `.java`.
6. 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).
7. 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.
    2. A file is unsaved if you see a white dot in the tab where the filename shows in VS Code
8. Fill in the items in the worksheet below (save as often as necessary).
9. Once finished, export the worksheet.
10. Add the output file to any location of your choice in your repository folder (i.e., a `Module2` folder).
11. Check that git sees it via `git status`.
12. If everything is good, continue to submit.

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

Branch name: M2-Java-Problems

Group

100%

Group: Problem 1
Tasks: 1
Points: 3

^ COLLAPSE ^

Task

100%

Group: Problem 1
Task #1: Screenshot of the Problem 1 Solved Code and Output
Weight: ~100%
Points: ~3.00

^ COLLAPSE ^

ⓘ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)

Columns: 1

Sub-Task

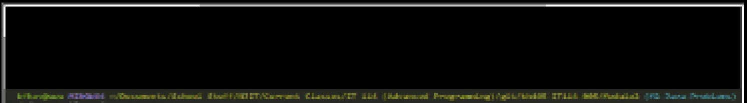
100%

Group: Problem 1
Task #1: Screenshot of the Problem 1 Solved Code and Output
Sub Task #1: Screenshot the output of the solved problem

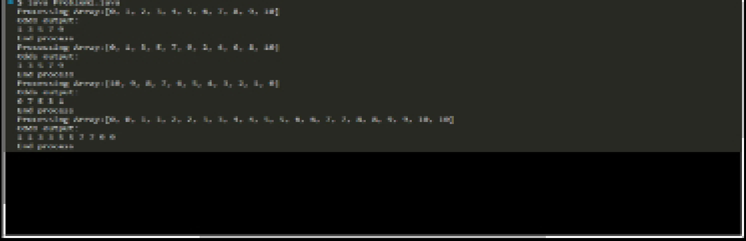
Task Screenshots

Gallery Style: 2 Columns

4 2 1



```
[illegible] ~/Documents/School Stuff/NJIT/Current Classes/IT 114 (Advanced Programming)/git/[illegible] IT114-006/Module2 (M2-Java-Problems)
```



```
$ java Problem1.java
Processing Array:[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Odds output:
1 3 5 7 9
End process
Processing Array:[0, 1, 3, 5, 7, 9, 2, 4, 6, 8, 10]
Odds output:
1 3 5 7 9
End process
Processing Array:[10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
Odds output:
9 7 5 3 1
End process
Processing Array:[0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10]
Odds output:
1 1 3 3 5 5 7 7 9 9
End process
```

Output displaying odd numbers from each array

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task

100%

Group: Problem 1

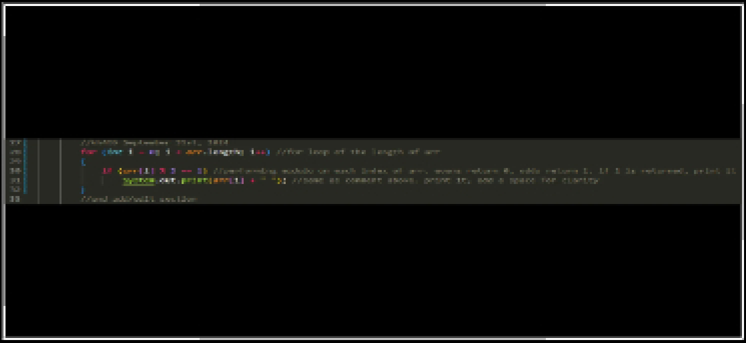
Task #1: Screenshot of the Problem 1 Solved Code and Output

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

Task Screenshots

Gallery Style: 2 Columns

4 2 1



```
        //[illegible] September 21st, 2024
        for (int i = 0; i < arr.length; i++) //for loop of the length of arr
        {
            if (arr[i] % 2 == 1) //performing modulo on each index of arr, evens return 0, odds return 1. If 1 is returned, print it
                System.out.print(arr[i] + " "); //same as comment above. print it, add a space for clarity
        }
        //end add/edit section
```

Code solution, with UCID and date included as a comment. Snippets of code explanation are also included in this screenshot.

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

The if statement, nested inside the for loop that is the length of arr, performs modulo by 2 of each index (int i, starting at 0). Modulo 2 on an even number returns 0, and returns 1 for an odd number. If 1 is returned, the index being evaluated is printed.

End of Task 1

End of Group: Problem 1
Task Status: 1/1

Group

Group: Problem 2

100%

Tasks: 1
Points: 3

COLLAPSE

Task

100%

Group: Problem 2
Task #1: Screenshot of the Problem 2 Solved Code and Output
Weight: ~100%
Points: ~3.00

COLLAPSE

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).

Columns: 1

Sub-Task

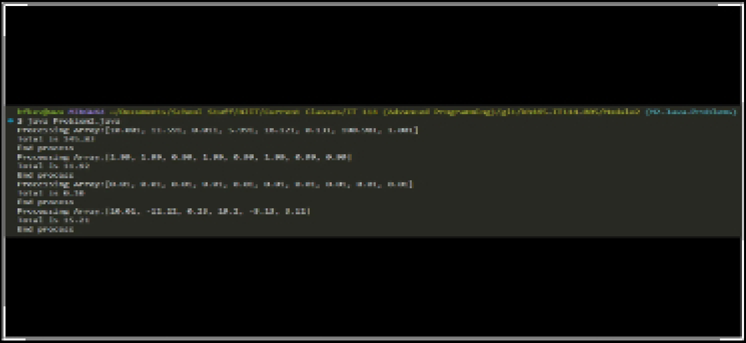
100%

Group: Problem 2
Task #1: Screenshot of the Problem 2 Solved Code and Output
Sub Task #1: Screenshot the output of the solved problem

Task Screenshots

Gallery Style: 2 Columns

4 2 1



```
[illegible] ~/Documents/School Stuff/NJIT/Current Classes/IT 114 (Advanced Programming)/git/[illegible]/Module2 (M2-Java-Problems)
$ java Problem2.java
Processing Array:[10.001, 11.591, 0.011, 5.991, 16.121, 0.131, 100.981, 1.001]
Total is 145.83
End process
Processing Array:[1.99, 1.99, 0.99, 1.99, 0.99, 1.99, 0.99, 0.99]
Total is 11.92
End process
Processing Array:[0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01]
Total is 0.10
End process
Processing Array:[10.01, -12.22, 0.23, 19.2, -5.13, 3.12]
Total is 15.21
End process
```

Output showing arrays being summed up properly, and having the solution rounded up to the nearest hundredth decimal

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task

100%

Group: Problem 2
Task #1: Screenshot of the Problem 2 Solved Code and Output
Sub Task #2: Screenshot the code solution (ucid/date must be included as a comment)

Task Screenshots

Gallery Style: 2 Columns

4 2 1

Code solution, with UCID and date included as a comment. Snippets of code explanation are also included in this screenshot.

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

The for loop iterates for the entire length of arr, with each loop adding the value of arr[i] to the variable total. Once this for loop is complete, the String totalOutput is formatted using String.format with %f specifying how to handle the floating point, and passing the variable total to it.

End of Task 1

End of Group: Problem 2
Task Status: 1/1

Group

100%

Group: Problem 3
Tasks: 1
Points: 3

COLLAPSE

Task

100%

Group: Problem 3
Task #1: Screenshot of the Problem 3 Solved Code and Output
Weight: ~100%
Points: ~3.00

COLLAPSE

Details:

Only make edits where the template code mentions

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.

Columns: 1

Sub-Task

100%

Group: Problem 3

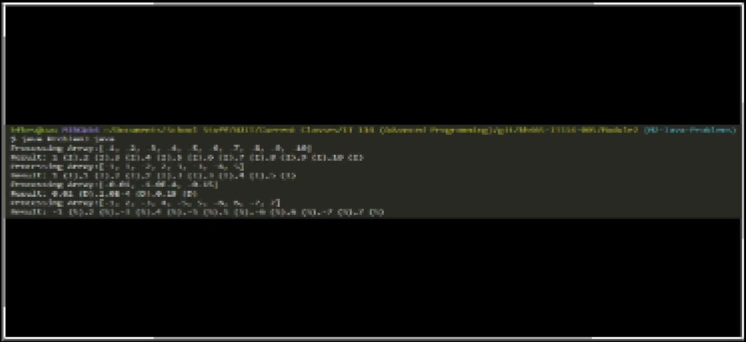
Task #1: Screenshot of the Problem 3 Solved Code and Output

Sub Task #1: Screenshot the output of the solved problem

Task Screenshots

Gallery Style: 2 Columns

4 2 1



```
[illegible]
$ java [illegible]
Processing Array:[illegible]
Result: [illegible]
Processing Array:[illegible]
Result: [illegible]
Processing Array:[illegible]
Result: [illegible]
Processing Array:[illegible]
Result: [illegible]
```

Output showing 3/4 arrays being converted to positive and keeping datatype. Could not get String array to work sadly.

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task

100%

Group: Problem 3

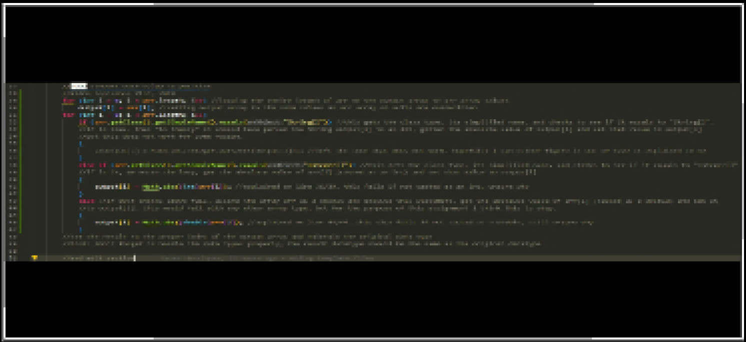
Task #1: Screenshot of the Problem 3 Solved Code and Output

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

Task Screenshots

Gallery Style: 2 Columns

4 2 1



Partial code solution, with UCID and date included as a comment. Other comments included in screenshot.

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

The first for loop loops through the length of arr and assigns the values of arr[i] to output[i]. The second for loop also loops through the length of arr, and has 3 conditions to check against. The first is if the simple name of the class arr equals "String[]", enter that if statement and set output[i] to the absolute value of output[i], parsed from a String to an int (though this did not work). If the first check fails, the second check checks to see if the simple name of the class arr equals "Integer[]". If so, enter the else if block, and set the value of output[i] to the absolute value of arr[i], casted as an int. Finally, if both checks fail, the else statement simply sets the value of output[i] to the absolute value of arr[i], casted as a double.

End of Task 1

End of Group: Problem 3
Task Status: 1/1

Group: Reflection
Tasks: 3
Points: 1

Group: Reflection
Task #1: Reflect on your experience
Weight: ~33%
Points: ~0.33

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.

Task Response Prompt

Response:

The main issue I had when attempting to solve the problem was trying to convert a String to an int, then back into a String. I could not get this to work, though I did attempt it in multiple ways. Another minor issue I had was needing to cast the values from Math.abs() to its respective datatype, but reading through the error and learning more about Math.abs() and casting, I was able to resolve that minor issue. Aside from that, I learned about getClass() and getSimpleName() from the problem itself by learning what they do, what they return, and how to use that returned name to compare two objects to each other.

End of Task 1

Task

100%

Group: Reflection
Task #2: Include the pull request link for this branch
Weight: ~33%
Points: ~0.33

COLLAPSE

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

Task URLs

URL #1
https://github.com/kh465/kh465-IT114-005/pull/6

URL
https://github.com/kh465/kh465-IT114-005/pull/(

URL #2
https://github.com/kh465/kh465-IT114-005/pull/5

URL
https://github.com/kh465/kh465-IT114-005/pull/!

End of Task 2

Task

100%

Group: Reflection
Task #3: Add Screenshot of Wakatime
Weight: ~33%
Points: ~0.33

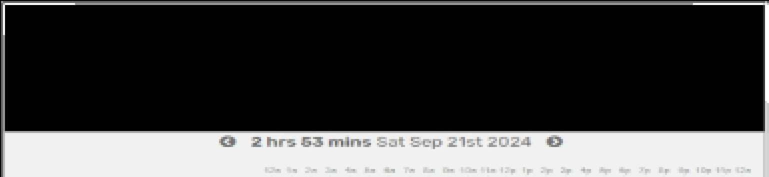
COLLAPSE

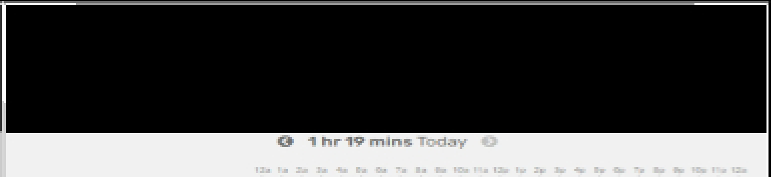
Details:
Note: The duration of time isn't directly related to the grade, the goal is to just make sure time is being tracked

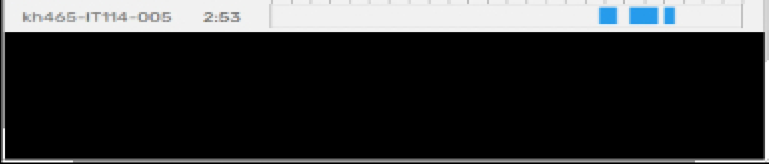
Task Screenshots

Gallery Style: 2 Columns

4 2 1

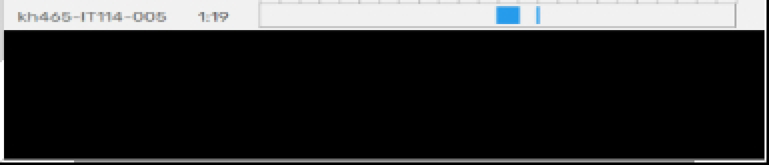






kh465-IT114-005 2:53

Screenshot of WakaTime on 9/21/24



kh465-IT114-005 1:19

Screenshot of WakaTime on 9/22/24

End of Task 3

End of Group: Reflection
Task Status: 3/3

End of Assignment