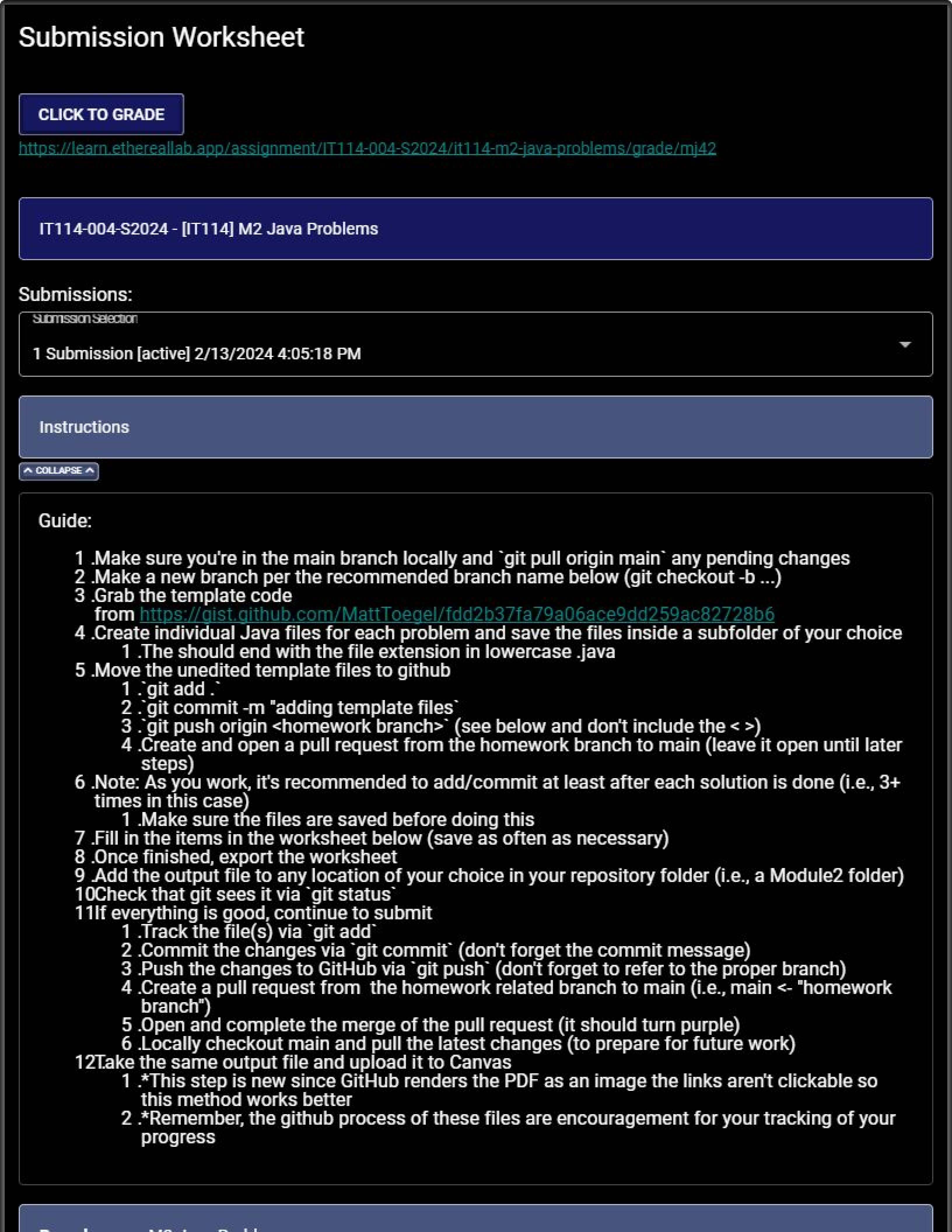


Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-004-S2024/it114-m2-java-problems/grade/mj42

IT114-004-S2024 - [IT114] M2 Java Problems

Submissions:

Submission Selection

1 Submission [active] 2/13/2024 4:05:18 PM

Instructions

COLLAPSE

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 .The 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 <homework branch>` (see below and don't include the < >)
    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)
10Check that git sees it via `git status`
11If 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)
12Take the same output file and upload it to Canvas
    1 .*This step is new since GitHub renders the PDF as an image the links aren't clickable so this method works better
    2 .*Remember, the github process of these files are encouragement for your tracking of your progress

Branch name: M2-Java-Problems

Tasks: 8 Points: 10.00

●
⌃ COLLAPSE ⌃

Problem 1 (3 pts.)

●
⌄ EXPAND ⌄

Task #1 - Points: 1

Text: Screenshot of the Problem 1 Solved Code and Output

●
⌄ EXPAND ⌄

Task #2 - Points: 1

Text: Explain your solution

●
⌃ COLLAPSE ⌃

Problem 2 (3 pts.)

●
⌄ EXPAND ⌄

Task #1 - Points: 1

Text: Screenshot of the Problem 2 Solved Code and Output

●
⌄ EXPAND ⌄

Task #2 - Points: 1

Text: Explain your solution

●
⌃ COLLAPSE ⌃

Problem 3 (3 pts.)

●
⌃ 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)

Checklist

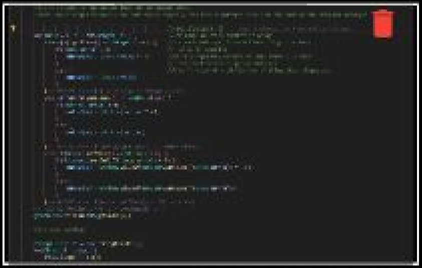
*The checkboxes are for your own tracking

#	Points	Details
□		Edits were done only in the bePositive() method and original template code/comments remain

☐ #1	1	untouched
☐ #2	1	Only arr is used (no direct usage of a1, a2, a3, a4)
☐ #3	5	Passed in array's values will get converted to a positive version AND converted back to the original data type
☐ #4	1	Includes code comments with student's ucid and date
☐ #5	1	Terminal output is fully visible

Task Screenshots:

Large Gallery



Checklist Items (0)

Checklist Items (0)

Missing Caption	Missing Caption

● COLLAPSE

Task #2 - Points: 1

Text: Explain your solution

Checklist *The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Clearly explains how the code/logic solves the problem (mentions both the conversion to positive and conversion to original data type)

Response:

The flow of the code like this. Go through every element in the array. Check the datatype and go into a unique conditional block depending on the type. Then convert the datatype if needed to an integer, multiplied by negative one, and then convert that value back to the original datatype.

● COLLAPSE

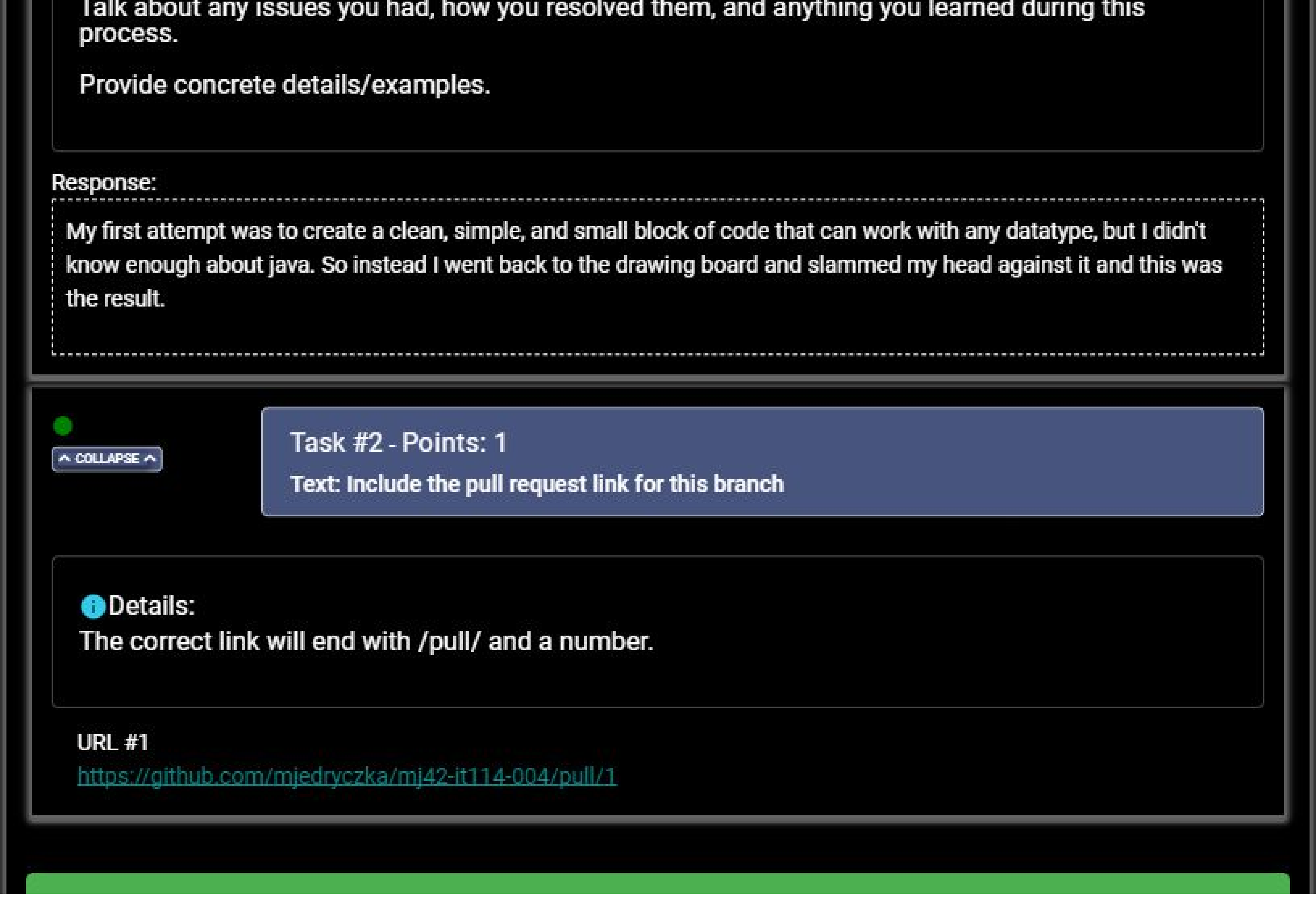
Reflection (1 pt.)

● 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.

Response:

My first attempt was to create a clean, simple, and small block of code that can work with any datatype, but I didn't know enough about java. So instead I went back to the drawing board and slammed my head against it and this was the result.

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/mjedryczka/mj42-it114-004/pull/1