Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-005-F2024/it114-module-2-java-problems/grade/bs679

Course: IT114-005-F2024
Assigment: [IT114] Module 2 Java Problems
Student: Brandon S. (bs679)

Submissions:

Submission Selection

1 Submission [submitted] 9/25/2024 2:27:17 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. 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

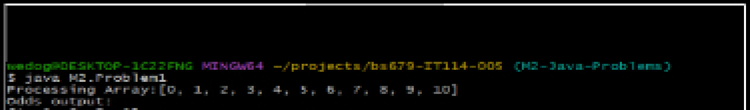
100%

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

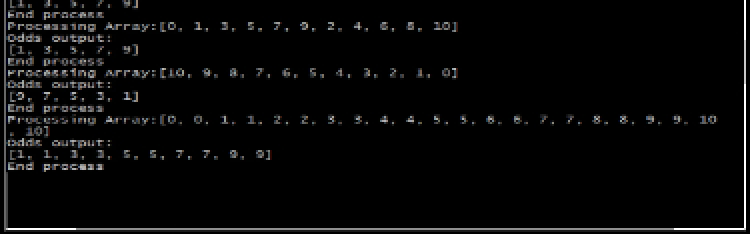
Task Screenshots

Gallery Style: 2 Columns

4 2 1



```
wedog@DESKTOP-1C22FNG MINGW64 ~/projects/bs679-IT114-005 (M2-Java-Problems)
$ java M2.Problem1
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 of problem 1

Caption(s) (required) ✓

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

Sub-Task

100%

Group: Problem 1

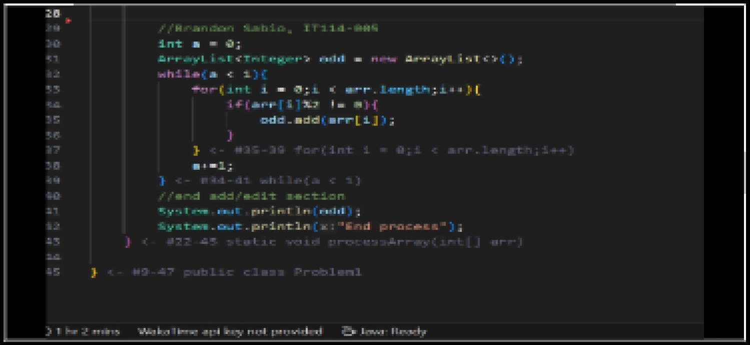
Task #1: Screenshot of the Problem 1 Solved Code and Output

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

Task Screenshots

Gallery Style: 2 Columns

4 2 1



```
        //Brandon Sabio, IT114-005
        int a = 0;
        ArrayList<Integer> odd = new ArrayList<>();
        while(a < 1){
            for(int i = 0;i < arr.length;i++){
                if(arr[i]%2 != 0){
                    odd.add(arr[i]);
                }
            } <- #35-39 for(int i = 0;i < arr.length;i++)
            a+=1;
        } <- #34-41 while(a < 1)
        //end add/edit section
        System.out.println(odd);
        System.out.println(x:"End process");
    } <- #22-45 static void processArray(int[] arr)

} <- #9-47 public class Problem1
```

code solution

Caption(s) (required) ✓

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

Task Response Prompt

Explain in concise steps how this logically works

Response:

In the block of code it creates an integer 'a' that keeps the while loop from running forever and an empty list that can be added to later. A while loop is created that says that if the integer in the array has a remainder, it gets added to the list 'odd'. Once the loop has gone through the array it adds to 'a' and ends the loop.

End of Task 1

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

Group

100%

Group: Problem 2
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

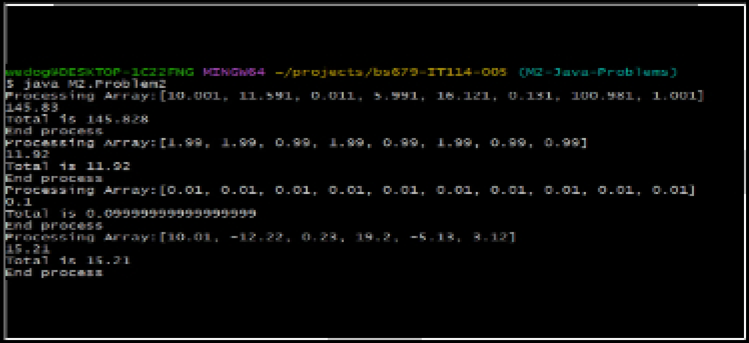
100%

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



```
wedog@DESKTOP-1C22FNG MINGW64 ~/projects/bs679-IT114-005 (M2-Java-Problems)
$ java M2.Problem2
Processing Array:[10.001, 11.591, 0.011, 5.991, 16.121, 0.131, 100.981, 1.001]
145.83
Total is 145.828
End process
Processing Array:[1.99, 1.99, 0.99, 1.99, 0.99, 1.99, 0.99, 0.99]
11.92
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]
0.1
Total is 0.09999999999999999
End process
Processing Array:[10.01, -12.22, 0.23, 19.2, -5.13, 3.12]
15.21
Total is 15.21
End process
```

problem 2 output

Caption(s) (required) ✓

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

Sub-Task

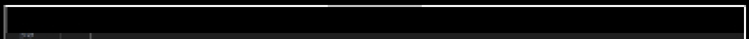
100%

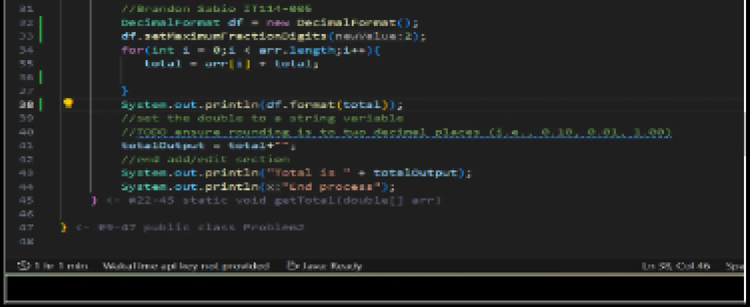
Group: Problem 2
Task #1: Screenshot of the Problem 2 Solved Code and Output
Sub Task #2: Screenshot the code solution (ucid/date must be included as a comment)

Task Screenshots

Gallery Style: 2 Columns

4 2 1





```
        //Brandon Sabio IT114-006
        DecimalFormat df = new DecimalFormat();
        df.setMaximumFractionDigits(newValue:2);
        for(int i = 0;i < arr.length;i++){
            total = arr[i] + total;

        }
        System.out.println(df.format(total));
        //set the double to a string variable
        //TODO ensure rounding is to two decimal places (i.e., 0.10, 0.01, 1.00)
        totalOutput = total+"";
        //end add/edit section
        System.out.println("Total is " + totalOutput);
        System.out.println(x:"End process");
    } <- #22-45 static void getTotal(double[] arr)

} <- #9-47 public class Problem2
```

code for problem 2

Caption(s) (required) ✓

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

Task Response Prompt

Explain in concise steps how this logically works

Response:

In this problem, I added all of the doubles in the array together in order to get the sum. To print them in the money format, I used decimal format so that only a maximum of 2 decimal places could be shwon.

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.

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

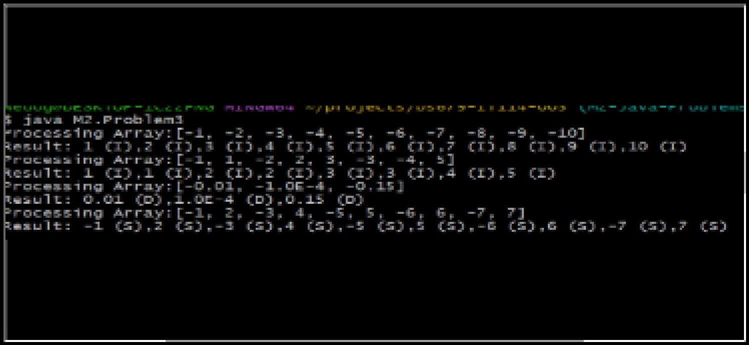
100%

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] MINGW64 ~/projects/[illegible] ([illegible])
$ java M2.Problem3
Processing Array:[-1, -2, -3, -4, -5, -6, -7, -8, -9, -10]
Result: 1 (I),2 (I),3 (I),4 (I),5 (I),6 (I),7 (I),8 (I),9 (I),10 (I)
Processing Array:[-1, 1, -2, 2, 3, -3, -4, 5]
Result: 1 (I),1 (I),2 (I),2 (I),3 (I),3 (I),4 (I),5 (I)
Processing Array:[-0.01, -1.0E-4, -0.15]
Result: 0.01 (D),1.0E-4 (D),0.15 (D)
Processing Array:[-1, 2, -3, 4, -5, 5, -6, 6, -7, 7]
Result: -1 (S),2 (S),-3 (S),4 (S),-5 (S),5 (S),-6 (S),6 (S),-7 (S),7 (S)
```

output for problem 3

Caption(s) (required) ✓

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

Sub-Task

100%

Group: Problem 3

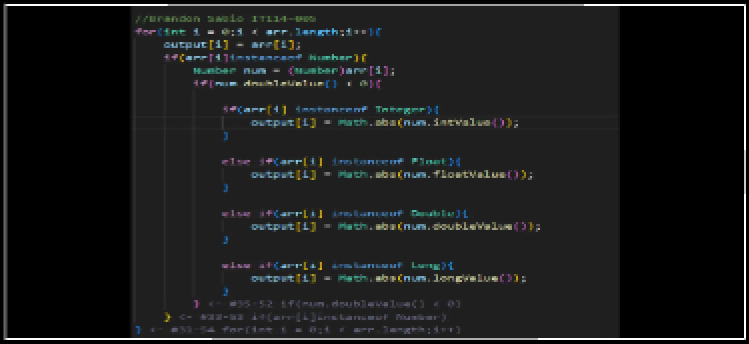
Task #1: Screenshot of the Problem 3 Solved Code and Output

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

Task Screenshots

Gallery Style: 2 Columns

4 2 1



```java
//Brandon Sasio IT114-005
for(int i = 0;i < arr.length;i++){
    output[i] = arr[i];
    if(arr[i]instanceof Number){
        Number num = (Number)arr[i];
        if(num.doubleValue() < 0){

            if(arr[i] instanceof Integer){
                output[i] = Math.abs(num.intValue());
            }

            else if(arr[i] instanceof Float){
                output[i] = Math.abs(num.floatValue());
            }

            else if(arr[i] instanceof Double){
                output[i] = Math.abs(num.doubleValue());
            }

            else if(arr[i] instanceof Long){
                output[i] = Math.abs(num.longValue());
            }
        } <- #35-52 if(num.doubleValue() < 0)
    } <- #33-53 if(arr[i]instanceof Number)
} <- #31-54 for(int i = 0;i < arr.length;i++)
```

code for problem 3

Caption(s) (required) ✓

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

Task Response Prompt

Explain in concise steps how this logically works

Response:

This program goes through the array and checks if there is a number. If there is, it will go through the line to check what data type it is and then convert the number into a positive number.

End of Task 1

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

Group

100%

Group: Reflection
Tasks: 3
Points: 1

^ COLLAPSE ^

Task

100%

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

^ COLLAPSE ^

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:

For this I had some problems, but it was mostly because I was never good at programming in java. For example, I struggled with formatting the decimals in problem 2. To try and figure it out, I used online resources and found a solution that I thought would help me solve the problem. Once I tried it out, it helped me solve the problem.

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/bsabio/bs679-IT114-005/pull/3

URL
https://github.com/bsabio/bs679-IT114-005/pull/

End of Task 2

Task

100%

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

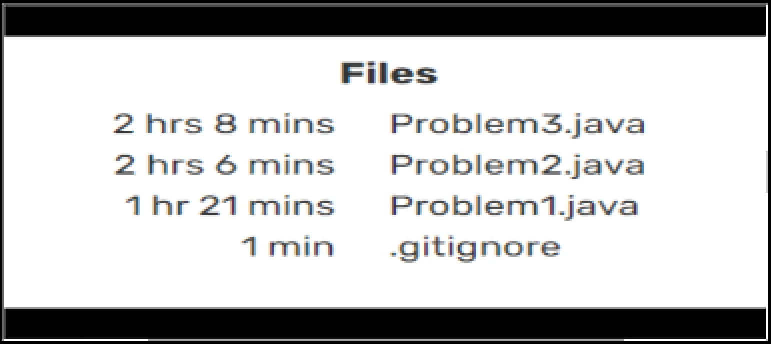
COLLAPSE

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



Files

2 hrs 8 mins	Problem3.java
2 hrs 6 mins	Problem2.java
1 hr 21 mins	Problem1.java
1 min	.gitignore

waka time

End of Task 3

End of Group: Reflection
Task Status: 3/3

End of Assignment