Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-006-S2024/it114-m2-java-problems/grade/sb57

IT114-006-S2024 - [IT114] M2 Java Problems

Submissions:

Submission Selection

1 Submission [active] 4/29/2024 8:55:12 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)
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
    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.)

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)

Checklist

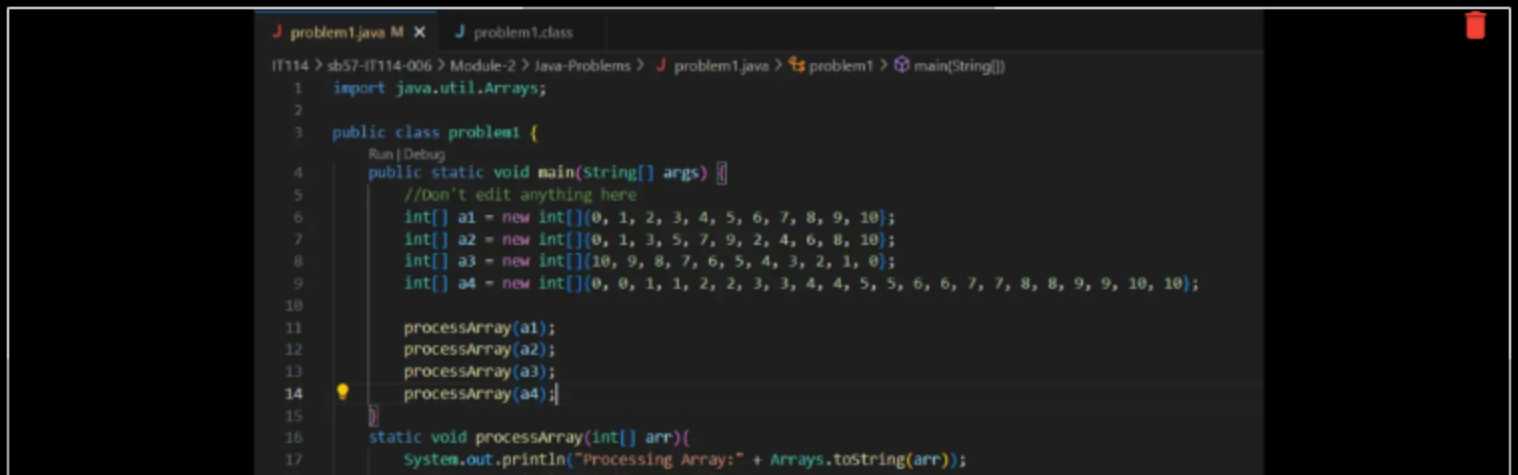
*The checkboxes are for your own tracking

#	Points	Details
#1	1	Edits were done only in the processArray() method and original template code/comments remain untouched
#2	1	Only arr is used (no direct usage of a1, a2, a3, a4)
#3	5	Only odd values output (not odd indexes/keys)
#4	1	Includes code comments with student's ucid and date
#5	1	Terminal output is fully visible

Task Screenshots:

Gallery Style: Large View

Small Medium Large



```java
        System.out.println(x:"Odds output:");
        //hint: use the arr variable; don't diretly use the a1-a4 variables
        //TODO add/edit code here

        //Shreya Bose
        //sb57
        //February 5, 2024

        for (int num : arr){
            if (num % 2 != 0){
                System.out.print(num + ", ");
            }
        }

        //end add/edit section
        System.out.println();
        System.out.println(x:"End process");
    }

}
```

Screenshot of code edits.

Checklist Items (4)

#1 Edits were done only in the processArray() method and original template code/comments remain untouched

#2 Only arr is used (no direct usage of a1, a2, a3, a4)

#3 Only odd values output (not odd indexes/keys)

#4 Includes code comments with student's ucid and date

```
PS C:\Users\Shreya\Desktop\IT114\sb57-IT114-006\Module-2\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:
 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
PS C:\Users\Shreya\Desktop\IT114\sb57-IT114-006\Module-2\Java-Problems>
```

Screenshot of complete terminal output.

Checklist Items (2)

#3 Only odd values output (not odd indexes/keys)

#5 Terminal output is fully visible

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 how the odd values are determined)

Response:

I used a loop to iterate through the arrays and only output the odd numbers. I determined the odd values by finding which values have a remainder when divisible by 2, and outputted them.

COLLAPSE

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

Checklist *The checkboxes are for your own tracking

#	Points	Details
#1	1	Edits were done only in the getTotal() method and original template code/comments remain untouched (unless noted)
#2	1	Only arr is used (no direct usage of a1, a2, a3, a4)
#3	5	Passed in array's values get summed AND rounded to two decimal places like currency (i.e., 0.00, 0.10, 1.10)
#4	1	Includes code comments with student's ucid and date
#5	1	Terminal output is fully visible

Task Screenshots:

Gallery Style: Large View

Small Medium Large

problem2.java

IT114 > sb57-IT114-006 > Module-2 > Java-Problems > problem2.java > problem2 > main(String[])

```java
import java.util.Arrays;

public class problem2 {
    Run | Debug
    public static void main(String[] args) {
        //Don't edit anything here
        double[] a1 = new double[]{10.001, 11.591, 0.011, 5.991, 16.121, 0.131, 100.981, 1.001};
        double[] a2 = new double[]{1.99, 1.99, 0.99, 1.99, 0.99, 1.99, 0.99, 0.99};
        double[] a3 = new double[]{0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01};
        double[] a4 = new double[]{10.01, -12.22, 0.23, 19.20, -5.13, 3.12};

        getTotal(a1);
        getTotal(a2);
        getTotal(a3);
        getTotal(a4);
    }
    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

        //Shreya Bose
        //sb57
        //February 5, 2024

        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)

        totalOutput = String.format(format:"%.2f", total);

        //end add/edit section
        System.out.println("Total is " + totalOutput);
        System.out.println(x:"End process");
    }

}
```

Screenshot of edited code.

Checklist Items (4)

#1 Edits were done only in the getTotal() method and original template code/comments remain untouched (unless noted)

#2 Only arr is used (no direct usage of a1, a2, a3, a4)

#3 Passed in array's values get summed AND rounded to two decimal places like currency (i.e., 0.00, 0.10, 1.10)

#4 Includes code comments with student's ucid and date

```
PS C:\Users\Shreya\Desktop\IT114\sb57-IT114-006\Module-2\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
PS C:\Users\Shreya\Desktop\IT114\sb57-IT114-006\Module-2\Java-Problems>
```

Screenshot of complete terminal output

Checklist Items (2)

#3 Passed in array's values get summed AND rounded to two decimal places like currency (i.e., 0.00, 0.10, 1.10)

#5 Terminal output is fully visible

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 how the values get summed and how the rounding is solved correctly)

Response:

I used a loop to iterate through the arrays and add each value to the total for each array starting the total value at 0. I also solved the formatting correctly to round to 2 decimals by suing String.format and %.2f.

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)

Requires at least 2 screenshots (code + output from terminal)

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Edits were done only in the bePositive() method and original template code/comments remain 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:

Gallery Style: Large View

Small Medium Large

problem3.java

IT114 > sb57-IT114-006 > Module-2 > Java-Problems > problem3.java > problem3 > bePositive(T[]) <T>

```
public class problem3 {
    public static void main(String[] args) {

        bePositive(a1);
        bePositive(a2);
        bePositive(a3);
        bePositive(a4);
    }
    // <T> turns this into a generic so it can take in any datatype, it'll be passed as an Object so casting is required
    static <T> void bePositive(T[] arr){
        System.out.println("Processing Array:" + Arrays.toString(arr));
        //your code should set the indexes of this array
        Object[] output = new Object[arr.length];
        //hint: use the arr variable; don't diretly use the a1-a4 variables
        //TODO convert each value to positive
        //set the result to the proper index of the output array
        //hint: don't forget to handle the data types properly, the result datatype should be the same as the original datatype

        //Shreya Bose
        //sb57
        //February 5, 2024

        for (int i = 0; i < arr.length; i++) {
            T element = arr[i];
            if (element instanceof Integer) {
                output[i] = Math.abs((Integer) element);
            } else if (element instanceof Double) {
                output[i] = Math.abs((Double) element);
            } else if (element instanceof String) {
                output[i] = Integer.toString(Math.abs(Integer.parseInt((String) element)));
            }
        }

        //end edit section

        StringBuilder sb = new StringBuilder();
        for(Object i : output){
            if(sb.length() > 0){
```

Screenshot of edited code. (Only edits shown, whole code does not fit in screenshot but did not edit anything else from template than what is shown)

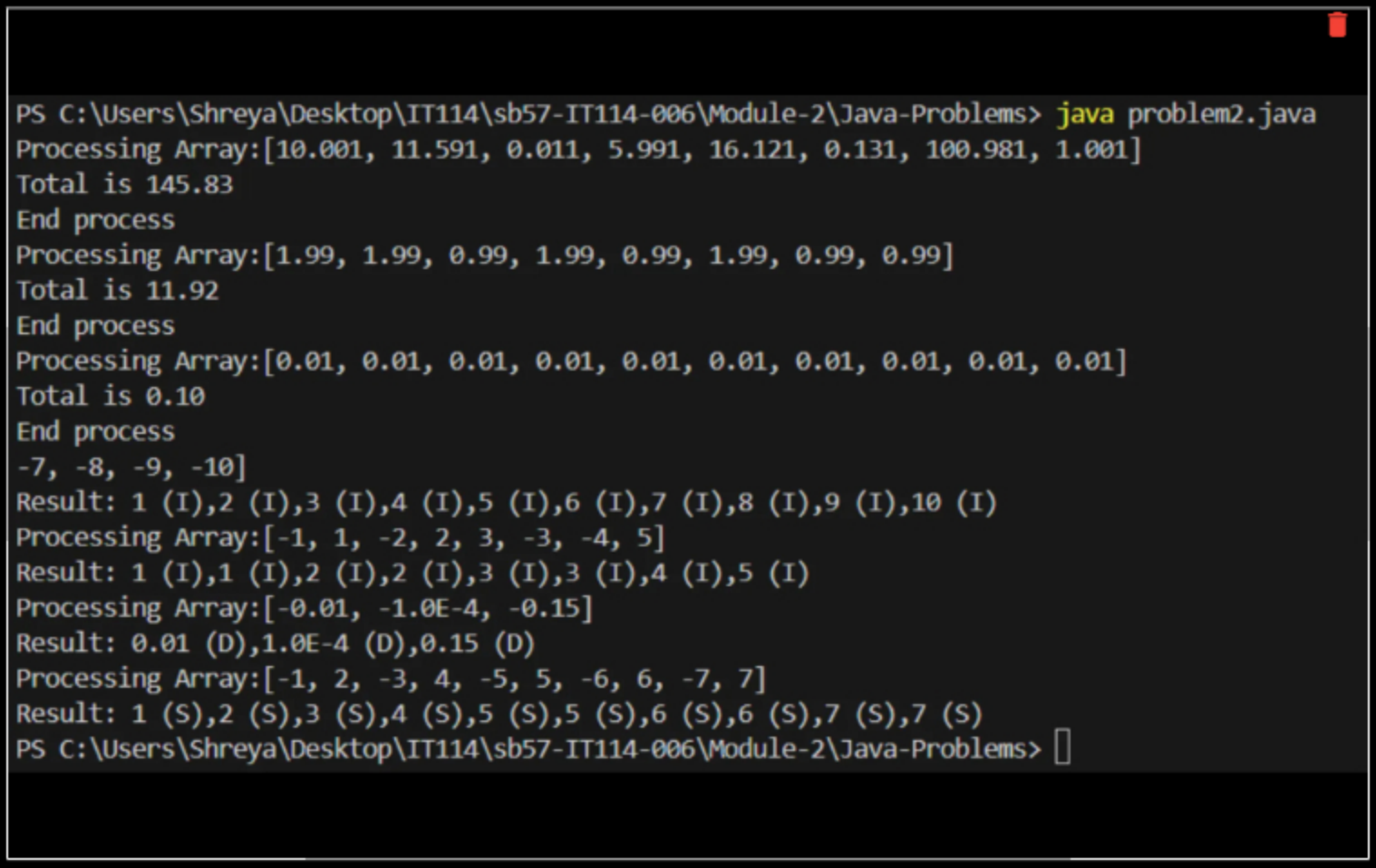
Checklist Items (4)

#1 Edits were done only in the bePositive() method and original template code/comments remain untouched

#2 Only arr is used (no direct usage of a1, a2, a3, a4)

#3 Passed in array's values will get converted to a positive version AND converted back to the original data type

#4 Includes code comments with student's ucid and date



Screenshot of complete terminal output.

Checklist Items (2)

#3 Passed in array's values will get converted to a positive version AND converted back to the original data type

#5 Terminal output is fully visible

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:

I use a loop to iterate through each array and use Math.abs() to output the positive value of each array value. I also use

I use a loop to iterate through each array and use Math.abs() to output the positive value of each array value. I also use an if/else if to convert it to the original data type.

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:

I ran into issues trying to compile my java file using vscode, since i usually use a different ide when working with java. However, I found a much simpler way to compile through the terminal instead.

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/sb57-shreya/sb57-IT114-006/pull/4

End of Assignment