

# Submission Worksheet

## Submission Data

**Course:** IT114-003-F2025

**Assignment:** IT114 Java Problems

**Student:** Rayyan K. (rk975)

**Status:** Submitted | **Worksheet Progress:** 100%

**Potential Grade:** 10.00/10.00 (100.00%)

**Received Grade:** 0.00/10.00 (0.00%)

**Started:** 9/27/2025 3:43:06 PM

**Updated:** 9/27/2025 7:31:05 PM

**Grading Link:** <https://learn.ethereallab.app/assignment/v3/IT114-003-F2025/it114-java-problems/grading/rk975>

**View Link:** <https://learn.ethereallab.app/assignment/v3/IT114-003-F2025/it114-java-problems/view/rk975>

## Instructions

- Overview Link: <https://youtu.be/Mrahk6SFYao>

1. Ensure you read all instructions and objectives before starting.
2. Create a new branch from `main` called `M2-Homework`
  1. `git checkout main` (ensure proper starting branch)
  2. `git pull origin main` (ensure history is up to date)
  3. `git checkout -b M2-Homework` (create and switch to branch)
3. Copy the template code from here: [GitHub Repository - M2 Homework](#)
  - It includes Problems 1-4 and a `BaseClass`. Put all into an `M2` folder or similar (adjust package reference at the top if you chose a different folder name).
  - Immediately record to history
    - `git add .`
    - `git commit -m "adding M2 HW baseline files"`
    - `git push origin M2-Homework`
    - Create a Pull Request from `M2-Homework` to `main` and keep it open
4. Fill out the below worksheet
  - Each Problem requires the following as you work
    - Ensure there's a comment with your UCID, date, and brief summary of how the problem was solved
    - Initial outline/plan of how you'll solve it via comments (add/commit after this stage)
    - Code solution (add/commit periodically as needed)
5. Once finished, click "Submit and Export"
6. Locally add the generated PDF to a folder of your choosing inside your repository folder and move it to Github
  1. `git add .`
  2. `git commit -m "adding PDF"`
  3. `git push origin M2-Homework`
  4. On Github merge the pull request from `M2-Homework` to `main`
7. Upload the same PDF to Canvas
8. Sync Local
  1. `git checkout main`
  2. `git merge M2-Homework`

# Section #1: ( 2 pts.) Problem 1 - Odds

Progress: 100%

≡ Task #1 ( 2 pts.) - Edit the `printOdds` method to output odd values of the array

Progress: 100%

## Details:

- Only make edits where noted via provided comments
- Challenge: Print odd values only in a single line separated by commas
- Step 1: sketch out plan using comments (include ucid and date)
- Step 2: Add/commit your outline of comments (required for full credit)
- Step 3: Add code to solve the problem (add/commit as needed)

## Part 1:

Progress: 100%

## Details:

Two screenshots are expected

- Snippet of relevant code showing solution (with ucid/date comment)
- Full output of executing the program

```
// Date: 9/27/25
// Solving problem solution steps:
// 1. Use a for loop to loop through the array
// 2. Check if number is odd by doing is & not equal to
// 3. If odd print out of the number followed by comma
// 4. Format will either be a trailing comma at end or need to add comma and get rid of
System.out.print("Output Array: ");
// Final Solution Below
public void printOdd() {
    boolean printFirst = true;
    for (int i = 0; i < arr.length; i++) {
        if (arr[i] % 2 != 0) {
            if (printFirst) {
                System.out.print(",");
            }
            System.out.print(arr[i]);
            printFirst = true;
        }
    }
}

// End Solution Below
System.out.println();
System.out.println();

}
public static void main(String[] args) {
    final String ucid = "rk975"; // --- change to your UCID
    // No edits below this line
}
```

Code-Date-Ucid

```
rkhan@Ray_Pro CLANGARM64 ~/IT1114/rk975-IT1114-003-2025 (M2-Homework)
$ Java M2/Problem1.java
Running Problem 1 for [rk975] [2025-09-27T15:39:30.742386600]
Objective: Print out only odd values in a single line separate by commas
Problem 1: Original Array: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
Output Array: 1,3,5,7,9

Problem 2: Original Array: [9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
Output Array: 9,7,5,3,1

Problem 3: Original Array: [0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9]
Output Array: 1,1,3,3,5,5,7,7,9,9

Problem 4: Original Array: [9, 9, 8, 8, 7, 7, 6, 6, 5, 5, 4, 4, 3, 3, 2, 2, 1, 1, 0, 0]
Output Array: 9,9,7,7,5,5,3,3,1,1

Completed Problem 1 for [rk975] [2025-09-27T15:39:30.768098600]
rkhan@Ray_Pro CLANGARM64 ~/IT1114/rk975-IT1114-003-2025 (M2-Homework)
$
```

Output display



Saved: 9/27/2025 3:51:24 PM

## ☞ Part 2:

Progress: 100%

### Details:

Direct link to the file in the homework related branch from Github (should end in `.java`)

URL #1

<https://github.com/rayk101/rk975>

IT114-~~100~~/~~2025~~Homework/M2/Problem1.java



URL

<https://github.com/rayk101/rk975>



Saved: 9/27/2025 3:51:24 PM

## ≡, Part 3:

Progress: 100%

### Details:

Briefly explain `how` the code solves the challenge (note: this isn't the same as `what` the code does)

Your Response:

The code solves the challenge by filtering out and printing only the odd numbers from the input array using a for loop, the code goes through the entire array. To properly structure the output with commas between integers and prevent a trailing comma at the end, I used a boolean flag. So that there is no comma at the end.



Saved: 9/27/2025 3:51:24 PM

## Section #2: ( 2 pts.) Problem 2 - Sum

Progress: 100%

≡ Task #1 ( 2 pts.) - Edit the `sumValues` method to sum the array values and present them in a format with exactly two decimal places

Progress: 100%

### Details:

- Only make edits where noted via provided comments
- Challenge 1: Sum all the values of the passed in array and assign to `total`
- Challenge 2: Have the sum be represented as a number with exactly 2 decimal
- Example: 0.1 would be shown as 0.10, 1 would be shown as 1.00, etc
- Step 1: sketch out plan using comments (include uid and date)

- Step 1: Create your planning comments (include date and desc)
- Step 2: Add/commit your outline of comments (required for full credit)
- Step 3: Add code to solve the problem (add/commit as needed)

## Part 1:

Progress: 100%

### Details:

Two screenshots are expected

- Snippet of relevant code showing solution (with uid/date comment)
- Full output of executing the program

```
// Challenge 1: Sum all the values of the passed array and assign to 'total'.
// Example: [1,2,3] would be stored in total & would be returned as 6.0000000000000005
// Keep all decimal and float using comments (class, function and doctests)
// Keep 2x add/commit your outline of comments (equivalent over 1000 lines)
// Add code and commit to solve this problem (commitments are required)
// 09/27/25
// Needs to return the double total with out the extra .0000000000000005
// 2. Add to the total sum
// 3. Add to the total sum
// 4. Format the string output to show 2 decimals by using string formatter (do not hardcode)
double total = 0;
// Solve challenge 1 here
// Solve challenge 2 here
for (double number : arr) {
    total += number;
}

// Solve Challenge 3 here
Killing multiThread = Killing();
Killing().start();
System.out.println("Raw Value: " + Killing().getRawValue());
System.out.println("Modified Value: " + Killing().getModifiedValue());
System.out.println("Total: " + Killing().getTotalValue());
}

public static void main(String[] args) {
    String uid = "rk975"; // Change to your UID
    String date = "09/27/25";
}
```

### code-solution

```
rkhargay@rk-OptiPlex-5070:~/Desktop/HW$ cd HW2 - M2-Homework (M2-Homework)
$ java M2/Problem2.java
Running Problem 2 for [1,2,3]
Objective: Print out the total sum of the passed array
Problem 2: Original Array: [1, 2, 3]
TOTAL Raw Value: 6.0
Total Modified Value: 6.00

Problem 3: Original Array: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
TOTAL Raw Value: 55.0
Total Modified Value: 55.00

Problem 4: Original Array: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
TOTAL Raw Value: 55.0
Total Modified Value: 55.00

Problem 5: Original Array: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
TOTAL Raw Value: 55.0
Total Modified Value: 55.00

Total Raw Value: 55.0
Total Modified Value: 55.00
```

### terminal-output



Saved: 9/27/2025 4:24:21 PM

## Part 2:

Progress: 100%

### Details:

Direct link to the file in the homework related branch from Github (should end in `.java`)

URL #1

<https://github.com/rayk101/rk975>

IT114-HW2025/homework/M2/Problem2.java



URL

<https://github.com/rayk101/rk975>



Saved: 9/27/2025 4:24:21 PM

**Part 3:**

Progress: 100%

**Details:**

Briefly explain `how` the code solves the challenges (note: this isn't the same as `what` the code does)

**Your Response:**

The code uses a for each loop to take the sum of all values in the array, makes sure every single element is included in the total. It then formats the result to exactly two decimal places using a string formatter(W3 schools). Making sure there are 2 decimal places after every sum.



Saved: 9/27/2025 4:24:21 PM

**Section #3: ( 2 pts.) Problem 3 - Conversion**

Progress: 100%

**Task #1 ( 2 pts.)** - Edit the `bePositive` method to make each value positive, convert it back to the orginal data type, and set it to the proper slot in the `output` array

Progress: 100%

**Details:**

- Only make edits where noted via provided comments
- Challenge 1: Make each value positive
- Challenge 2: Convert the values back to their original data type and assign it to the proper slot of the `output` array
- Step 1: sketch out plan using comments (include ucid and date)
- Step 2: Add/commit your outline of comments (required for full credit)
- Step 3: Add code to solve the problem (add/commit as needed)

**Part 1:**

Progress: 100%

**Details:**

Two screenshots are expected

1. Snippet of relevant code showing solution (with ucid/date comment)
2. Full output of executing the program



```
public class Problem3 {
    public static void main(String[] args) {
        int[] arr = {-1, 2, -3, 4, -5, 6, -7, 8, -9, 10};
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] < 0) {
                arr[i] = arr[i] * -1;
            }
        }
        System.out.println("Original Array: " + arr);
        System.out.println("Modified Array: " + arr);
    }
}
```

### code-solution

```
rayk101@rayk101-OptiPlex-5090:~/Desktop$ cd M2-Homework
rayk101@rayk101-OptiPlex-5090:~/Desktop/M2-Homework$ java Problem3
Original Array: [-1, 2, -3, 4, -5, 6, -7, 8, -9, 10]
Modified Array: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
rayk101@rayk101-OptiPlex-5090:~/Desktop/M2-Homework$
```

### terminal-output



Saved: 9/27/2025 6:07:40 PM

## ☞ Part 2:

Progress: 100%

#### Details:

Direct link to the file in the homework related branch from Github (should end in `.java`)

URL #1

<https://github.com/rayk101/rk975->

IT114\_100-2025/Homework/M2/Problem3.java



URL

<https://github.com/rayk101/rk975>



Saved: 9/27/2025 6:07:40 PM

## ☞ Part 3:

Progress: 100%

#### Details:

Briefly explain `how` the code solves the challenges (note: this isn't the same as `what` the code does)

Your Response:

The code solves the challenge by using a for loop to iterate over each element in the array. Then I used a bunch of if statements that evaluated if the data type of the element was correct using the builtin `instanceOf`. While using `Math.abs` to make all elements positive. The code keeps all the values the same and them all in the order they were.



Saved: 9/27/2025 6:07:40 PM

# Section #4: ( 2 pts.) Problem 4 - Strings

Progress: 50%

≡ Task #1 ( 2 pts.) - Edit the `transformText` method to solve the challenges

Progress: 100%

## Details:

- Only make edits where noted via provided comments
- Challenge 1: Remove non-alphanumeric characters except spaces
- Challenge 2: Convert text to Title Case
- Challenge 3: Trim leading/trailing spaces and remove duplicate spaces
- Result 1-3: Assign final phrase to `placeholderForModifiedPhrase`
- Step 1: sketch out plan using comments (include ucid and date)
- Step 2: Add/commit your outline of comments (required for full credit)
- Step 3: Add code to solve the problem (add/commit as needed)

## Part 1:

Progress: 100%

## Details:

Two screenshots are expected

- Snippet of relevant code showing solution (with ucid/date comment)
- Full output of executing the program

```
private static String transformText(String phrase) {
    // Remove non-alphanumeric characters except spaces
    phrase = phrase.replaceAll("[^\\w\\s]", "");
    // Convert text to Title Case
    phrase = phrase.replaceAll("\\b([a-z])\\b", "$1".toUpperCase());
    // Trim leading/trailing spaces and remove duplicate spaces
    phrase = phrase.trim();
    return phrase;
}

String placeholderForModifiedPhrase = "";
```

code-solution

```
rkhhan@Ray_Pro CLANGARM64 ~/IT114/rk975-IT114-003-2025 (M2-Homework)
$ Java M2/Problem4.java
Index[1] "Java Programming" | Middle: ""
Index[2] "Short" | Middle: ""
Index[3] "A" | Middle: ""
Index[4] "Even" | Middle: ""

Completed Problem 4 for [rkh975] [2025-09-27T19:13:07.745Z]
rkhhan@Ray_Pro CLANGARM64 ~/IT114/rk975-IT114-003-2025 (M2-Homework)
$
```

## terminal-output



Saved: 9/27/2025 7:20:54 PM

### ☞ Part 2:

Progress: 100%

#### Details:

Direct link to the file in the homework related branch from Github (should end in `.java`)

#### URL #1

<https://github.com/rayk101/rk975->

IT114-~~108~~-M25/homework/M2/Problem4.java



URL

<https://github.com/rayk101/rk975->



Saved: 9/27/2025 7:20:54 PM

### ≡, Part 3:

Progress: 100%

#### Details:

Briefly explain `how` the code solves the challenges (note: this isn't the same as `what` the code does)

#### Your Response:

To guarantee good formatting, the code first eliminates all non-alphanumeric characters from each string but not the spaces, after it trims and normalizes the spacing. It rebuilds the phrase with appropriate space after converting each word to Title Case by capitalizing the initial letter and lowercasing the remaining letters. All three of the main tasks are met when this final result is saved in `placeholderForModifiedPhrase`.



Saved: 9/27/2025 7:20:54 PM

## ☰ Task #2 (+ 1.11 pts.) - Edit the `transformText` method to solve the extra credit challenge (challenge 4)

Progress: 0%

#### Details:

- Only make edits where noted via provided comments
- Challenge 4: Extract middle 3 characters (beginning starts at middle of phrase)
- Assign result to 'placeholderForMiddleCharacters'
- If not enough characters assign "Not enough characters"

- Step 1: sketch out plan using comments (include ucid and date)
- Step 2: Add/commit your outline of comments (required for full credit)
- Step 3: Add code to solve the problem (add/commit as needed)

## ❑ Part 1:

Progress: 0%

### Details:

Two screenshots are expected

1. Snippet of relevant code showing solution (with ucid/date comment)
2. Full output of executing the program



Missing Caption



Saved: 9/16/2025 12:25:49 PM

## ≡ Part 2:

Progress: 0%

### Details:

Briefly explain `how` the code solves the extra credit challenge (note: this isn't the same as `what` the code does)

Your Response:

Missing Response



Saved: 9/16/2025 12:25:49 PM

# Section #5: ( 2 pts.) Misc

Progress: 100%

## ≡ Task #1 ( 0.67 pts.) - Github Details

Progress: 100%

## ❑ Part 1:

Progress: 100%

## Details:

From the Commits tab of the Pull Request screenshot the commit history Following minimum should be present

history

morehistory



Saved: 9/27/2025 7:26:32 PM

⊕ Part 2:

Progress: 100%

#### Details:

Include the link to the Pull Request (should end in `/pull/#`)

URI #1

<https://github.com/rayk101/rk975->

IT114-003-2025/



UR

<https://github.com/rayk101/rk975>



Saved: 9/27/2025 7:26:32 PM

Task #2 ( 0.67 pts.) - WakaTime - Activity

Progress: 100%

#### **Details:**

- Visit the WakaTime.com Dashboard

- Click **Projects** and find your repository
- Capture the overall time at the top that includes the repository name
- Capture the individual time at the bottom that includes the file time
- Note: The duration isn't relevant for the grade and the visual graphs aren't necessary



times



full overall



Saved: 9/27/2025 7:28:02 PM

## ≡ Task #3 ( 0.67 pts.) - Reflection

Progress: 100%

## ⇒ Task #1 ( 0.33 pts.) - What did you learn?

Progress: 100%

### Details:

Briefly answer the question (at least a few decent sentences)

### Your Response:

I discovered how to use simple Java functions like replaceAll, trim, and loops to clean and alter jumbled strings by eliminating special characters, cutting out unneeded spaces, and changing text to Title Case. In order to prevent compilation mistakes, I also realized how crucial it is to keep consistent logic and match variable names exactly.



Saved: 9/27/2025 7:29:58 PM

## ⇒ Task #2 ( 0.33 pts.) - What was the easiest part of the assignment?

Progress: 100%

### Details:

Briefly answer the question (at least a few decent sentences)

Your Response:

The easiest part was using replaceAll("[^a-zA-Z0-9 ]", "") to remove non-alphanumeric characters while keeping spaces. That one line handled a big chunk of the cleanup and made the rest of the formatting much easier.



Saved: 9/27/2025 7:30:34 PM

## ⇒ Task #3 ( 0.33 pts.) - What was the hardest part of the assignment?

Progress: 100%

### Details:

Briefly answer the question (at least a few decent sentences)

Your Response:

The hardest part was debugging mismatched variable names like TextCase vs Textcase, and making sure the string was properly cleaned before splitting and formatting. Small typos caused multiple compile errors, and fixing them required careful attention to naming and order of operations.



Saved: 9/27/2025 7:31:05 PM