# Submission Worksheet

## Submission Data

**Course:** IT114-450-M2025
**Assignment:** IT114 Module 3 User Input Challenges
**Student:** Mukaddis I. (mi348)
**Status:** Submitted | **Worksheet Progress:** 100%
**Potential Grade:** 10.00/10.00 (100.00%)
**Received Grade:** 0.00/10.00 (0.00%)
**Started:** 6/15/2025 2:04:51 PM
**Updated:** 6/15/2025 3:12:42 PM
**Grading Link:** https://learn.ethereallab.app/assignment/v3/IT114-450-M2025/it114-module-3-user-input-challenges/grading/mi348
**View Link:** https://learn.ethereallab.app/assignment/v3/IT114-450-M2025/it114-module-3-user-input-challenges/view/mi348

## Instructions

- Overview Link: https://youtu.be/iowHMCKuj5o

1. Ensure you read all instructions and objectives before starting.
2. Create a new branch from `main` called `M3-Homework`
    1. `git checkout main` (ensure proper starting branch)
    2. `git pull origin main` (ensure history is up to date)
    3. `git checkout -b M3-Homework` (create and switch to branch)
3. Copy the template code from here: GitHub Repository - M3 Homework
    - It includes CommandLineCalculator, SlashCommandHandler, MadLibsGenerator, a BaseClass and a stories folder with 5 stories (used for MadLibsGenerator). Put all into an `M3` 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 M3 HW baseline files"`
        - `git push origin M3-Homework`
        - Create a Pull Request from `M3-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
        - Update the `ucid` variable
        - 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 M3-Homework`
    4. On Github merge the pull request from `M3-Homework` to `main`

7. Upload the same PDF to Canvas
8. Sync Local
    1. `git checkout main`
    2. `git pull origin main`

# Section #1: ( 3 pts.) Challenge 1 - Command Line Calculator (Add/sub)

Progress: 100%

## ≔ Task #1 ( 3 pts.) - Edit the `main` method to solve the requirements

Progress: 100%

**Details:**

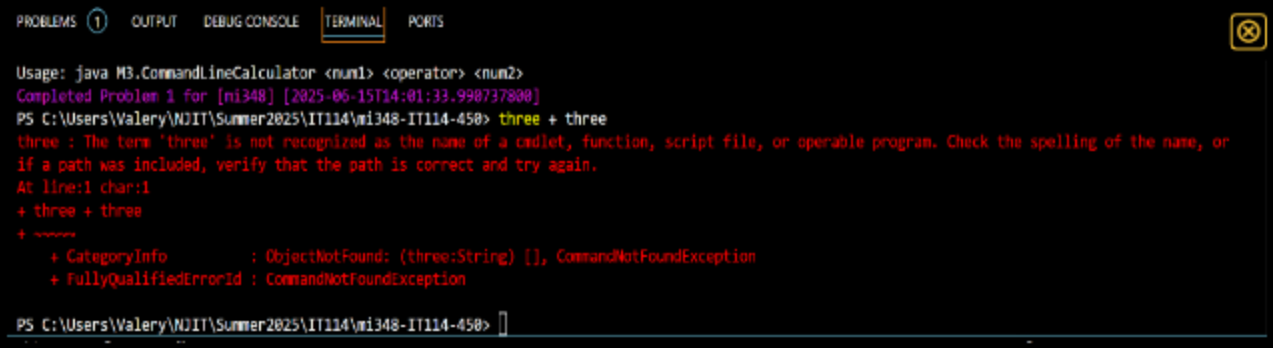
- Don't adjust the give code unless noted
- Challenge 1: Accept two numbers and an operator as command-line arguments (+ and -)
- Challenge 2: Allow integer and floating-point numbers
    - Ensure correct decimal places in output based on input (e.g., 0.1 + 0.2 → 1 decimal place)
- Display an error for invalid inputs or unsupported operators
- Add code to solve the problem (add/commit as needed)

## 🖼 Part 1:

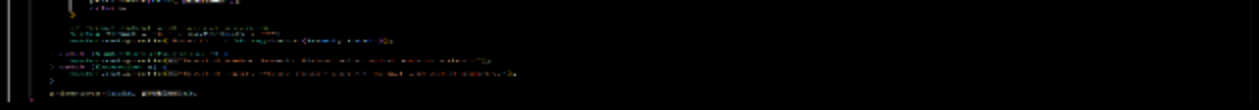
Progress: 100%

**Details:**
Two screenshots are expected

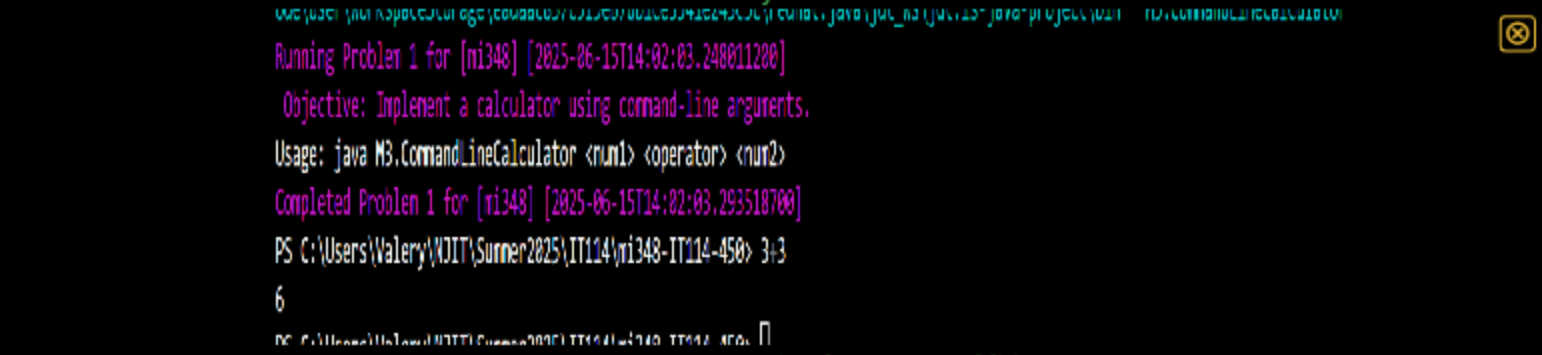
1. Snippet of relevant code showing solution (with ucid/date comment)
2. Full output of executing the program (Capture 5 variations of tests)



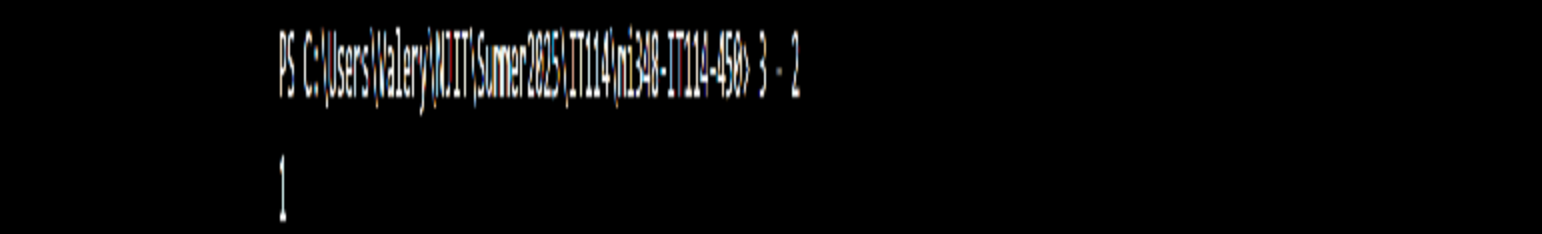

Error when using Strings



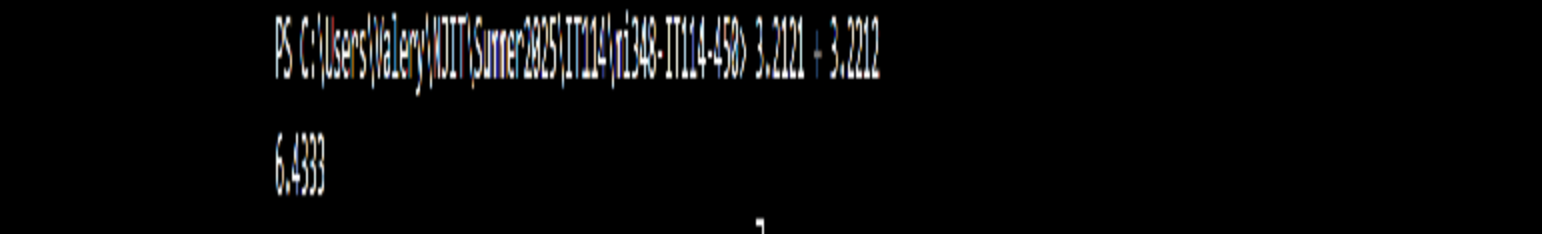
My cod with UCID and Date



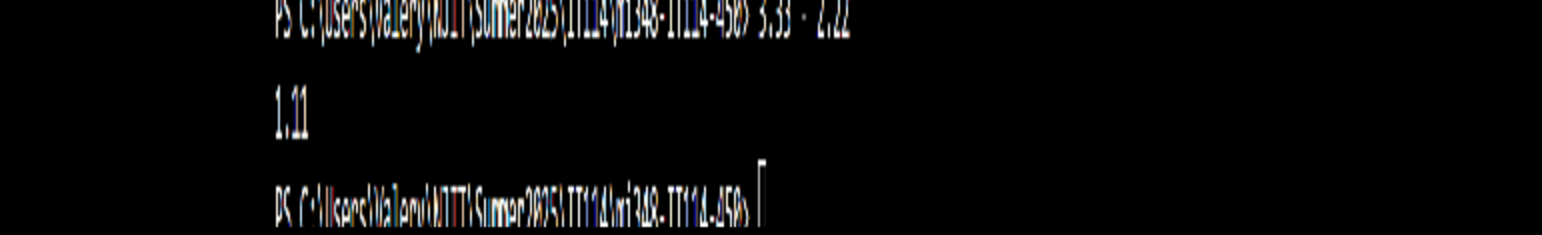

Simple Integer Addition


Subtracting Integers


Adding 2 floats




Subtracting 2 floats

Saved: 6/15/2025 2:47:51 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/MIsmail215/mi348-IT114-450/M3-Homework/IT114-2025-Module3-Homework/IT114-2025-Module3-Homework/M3/CommandLineCalculator.java

https://github.com/MIsmail215/m

Saved: 6/15/2025 2:47:51 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:

This program takes two numbers and an operator (+ or -) from the command line, does the math, and prints the result. It checks how many decimal places were used in the input and makes sure the answer is shown with the same level of detail. If the user types something wrong or uses an unsupported operator, it shows an error message.

Saved: 6/15/2025 2:47:51 PM

# Section #2: ( 3 pts.) Challenge 2 - Slash Command Handler

Progress: 100%

## Task #1 ( 3 pts.) - Edit the `main` method to solve the requirements

Progress: 100%

**Details:**

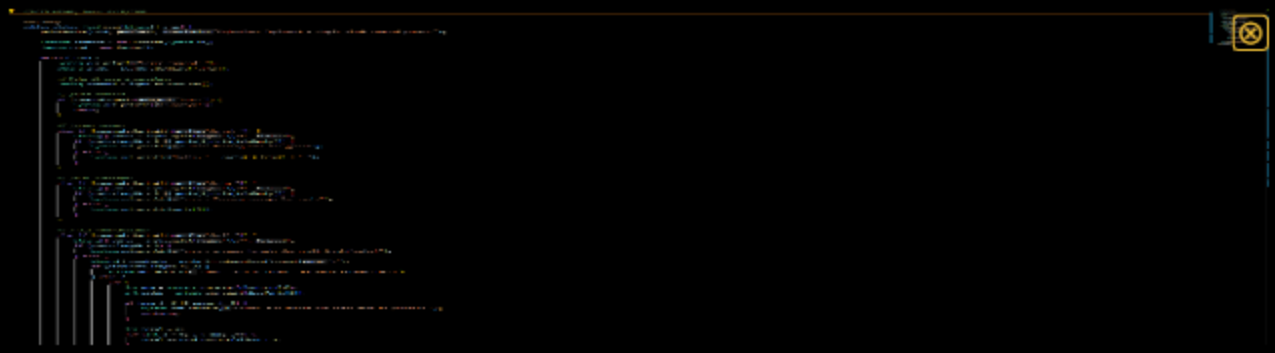
- Don't adjust the give code unless noted
- Challenge 1: Accept user input as slash commands (Commands are case-insensitive)
  - `"/greet <name>" → Prints "Hello, <name>!"`
  - `"/roll <num>d<sides>" → Roll <num> dice with <sides> and returns a`
  - `"/echo <message>" → Prints the message back`
  - `"/quit" → Exits the program`
- Challenge 2: Print an error for unrecognized commands
- Challenge 3: Print errors for invalid command formats (when applicable)
- 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 (Capture 3 variations of each command except "/quit")



Shows my code and UCID/Date

```
Enter command: /greet Mukaddis
Hello, Mukaddis!
Enter command: /greet Ismail
Hello, Ismail!
Enter command: /greet Jaden
Hello, Jaden!
Enter command:
```

Greet Command 3 times

```
Enter command: /echo HELLO
HELLO
Enter command: /echo GOODBYE
GOODBYE
Enter command: /echo YOOYOYOYOYOYOY
YOOYOYOYOYOYOY
Enter command:
```

Echo Command 3 times

```
Enter command: /roll 2d2
Rolled 2d2 and got 3!
Enter command: /roll 3d3
Rolled 3d3 and got 4!
Enter command: /roll 5d5
Rolled 5d5 and got 17!
Enter command:
```

Roll Command 3 times

```
Enter command: /quit

Goodbye!
```

Quit Command

Saved: 6/15/2025 2:48:07 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/MIsmail215/mi348-IT114-450/M3-Homework/IT114-2025-Module3-Homework/IT114-2025-Module3-Homework/M3/SlashCommandHandler.java

URL
https://github.com/MIsmail215/m

Saved: 6/15/2025 2:48:07 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 takes user input and checks if it matches one of the supported slash commands like /t, /roll, /echo, or /quit. It handles each command by splitting the input, validating the format, and printing the appropriate response. If the command is unrecognized or has an invalid format, it shows an error message.

Saved: 6/15/2025 2:48:07 PM

# Section #3: ( 3 pts.) Challenge 3 - Mad Libs Generator

Progress: 100%

## Task #1 ( 3 pts.) - Edit the `main` method to solve the challenges

Progress: 100%

**Details:**

- Don't adjust the give code unless noted
- Ensure you have the `stories` folder with the 5 stories
- Challenge 1: Load a **random** story from the "stories" folder
- Challenge 2: Extract **each line** into a collection (i.e., ArrayList)
- Challenge 3: Prompts user for each placeholder (i.e., `<adjective>`)
    - Any word the user types is acceptable, no need to verify if it matches the placeholder type
    - Any placeholder with underscores should display with spaces instead
- Challenge 4: Replace placeholders with user input (assign back to original slot in collection)
- 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 (Capture the process for at least 2 stories)

```
Enter a(n) adjective: Big
Enter a(n) animal: Bear
Enter a(n) verb ending in ing: Dancing
Enter a(n) place: Dubai
Enter a(n) food: Pizza
Enter a(n) object: Hammer
Enter a(n) adjective: Fat
Enter a(n) souvenir: Snow Globe

Your Completed Mad Libs Story:

Today, I went to the zoo and saw a Big Bear.
It was Dancing near the Dubai.
The zookeeper said it loved eating Pizza and sleeping under a Hammer.
Before I left, I bought a Fat Snow Globe from the gift shop!
```

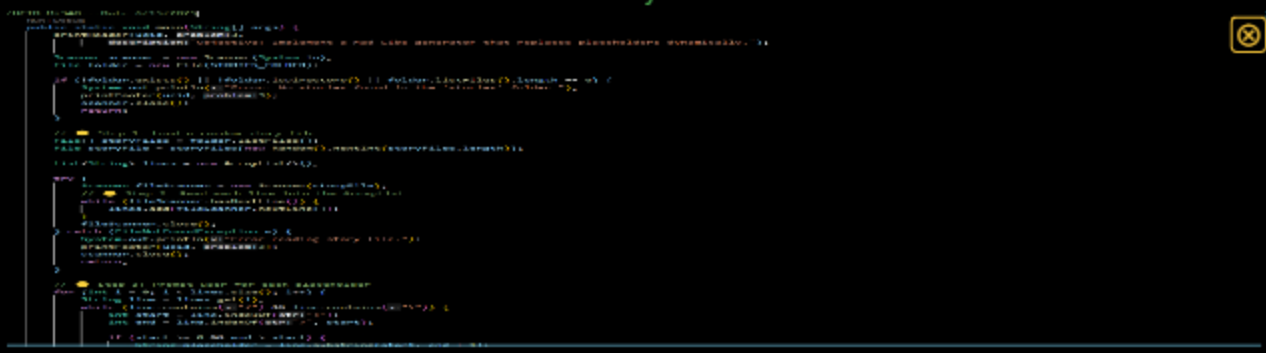
Story 1

```
Enter a(n) adjective: Small
Enter a(n) adjective: Run
Enter a(n) object: Vase
Enter a(n) adjective: Walk
Enter a(n) verb ending in ing: Running
Enter a(n) adjective: Short

Your Completed Mad Libs Story:

A Small witch gave me a potion that would make me Run.
She told me to drink it while standing on a Vase under the Walk moon.
As soon as I drank it, I started Running uncontrollably.
From that day forward, I became the most Short person in town.
```

Story 2



CODE UCID/Date

Saved: 6/15/2025 2:48:18 PM

Saved: 6/15/2025 2:48:18 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/MIsmail215/mi348-IT114-450/M3-Homework/IT114-2025-Module3-Homework/IT114-2025-Module3-Homework/M3/MadLibsGenerator.java

URL
https://github.com/MIsmail215/m

Saved: 6/15/2025 2:48:18 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 program picks a random story file from the "stories" folder and loads its lines into a list. It looks for placeholders like noun or verb_past_tense in each line and asks the user to enter a word to replace them. Once all placeholders are filled, the program prints out the final, completed story.

Saved: 6/15/2025 2:48:18 PM

# Section #4: ( 1 pt.) Misc

Progress: 100%

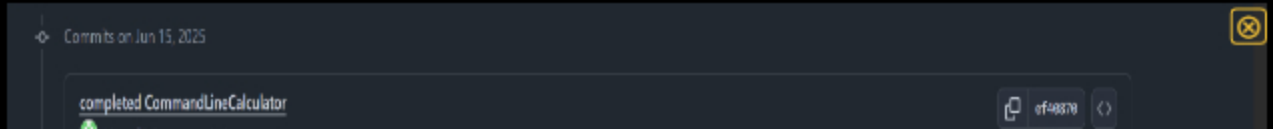
## Task #1 ( 0.33 pts.) - Github Details

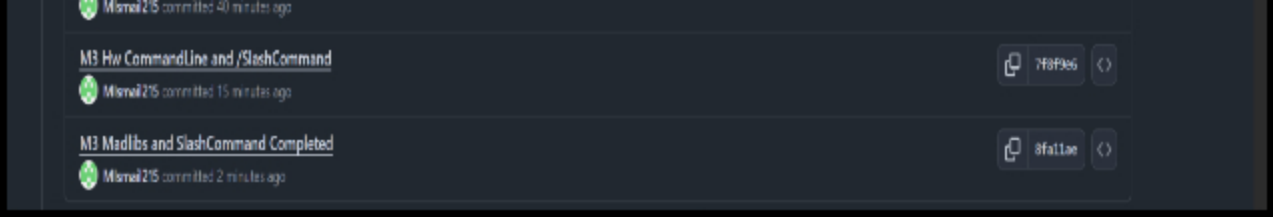
Progress: 100%

### Part 1:

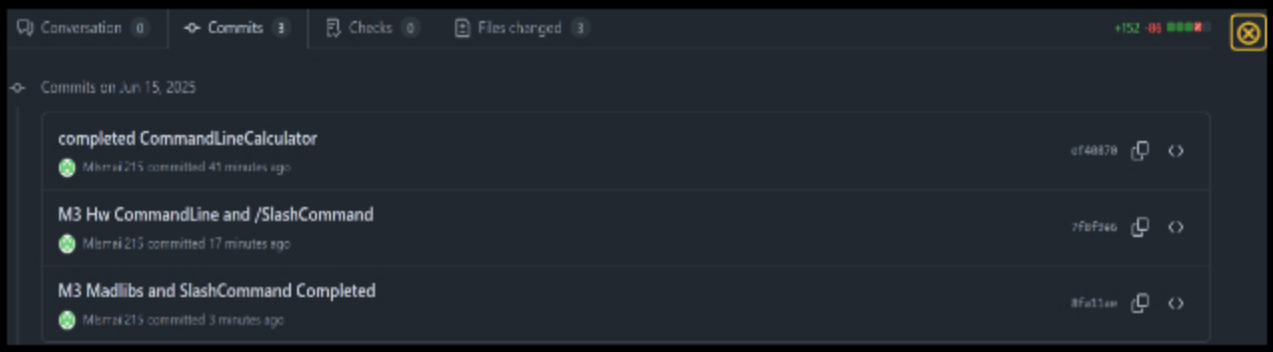
Progress: 100%

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






3 comments




Commit tab

Saved: 6/15/2025 2:47:02 PM

## Part 2:

Progress: 100%

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

URL #1
https://github.com/MIsmail215/mi348-IT11[illegible]

URL
https://github.com/MIsmail215/m

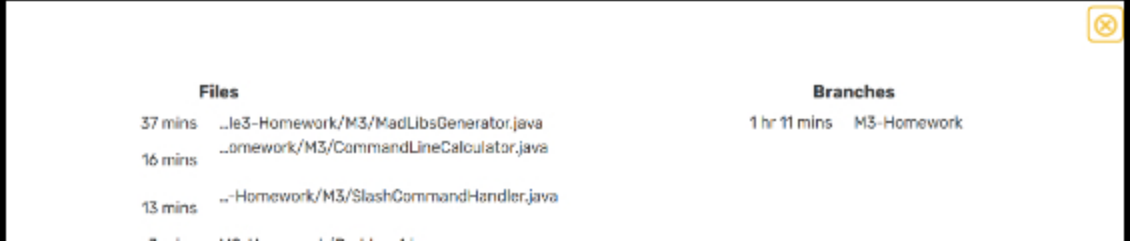
Saved: 6/15/2025 2:47:02 PM

# Task #2 ( 0.33 pts.) - WakaTime - Activity

Progress: 100%

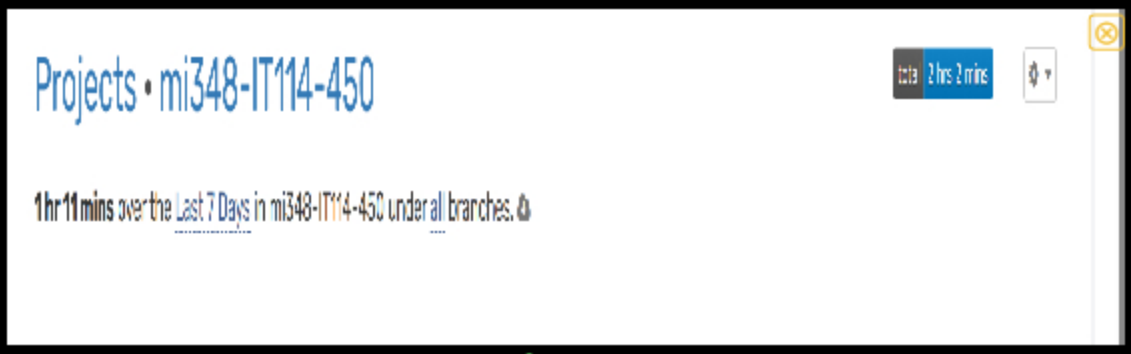
**Details:**

- 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



3 mins M2-Homework/Problem4.java
1 sec ...25-Module3-Homework/M3/BaseClass.java

File Names




Top of WAKATIME

Saved: 6/15/2025 3:06:38 PM

## Task #3 ( 0.33 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:

When using a Scanner to read input from System.in in Java, you should avoid closing the scanner if you plan to read input again later, because closing it will also close System.in, making further input impossible. It's best to create one scanner at the beginning of your program and reuse it, instead of creating multiple scanners for the same input stream. I also Learned you should avoid using it with System.in unless your program is about to end. Last thing I learned was only call scanner.close() when you're completely done reading input for the entire program.

Saved: 6/15/2025 3:10:22 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 of the assignment was utilizing user input with the Scanner class. It was pretty straightforward to prompt the user, read their input using nextLine() or nextInt(), and then display the result.

Saved: 6/15/2025 3:11:29 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 of the assignment was replacing placeholders in the Mad Libs Generator. It was tricky to locate each placeholder, prompt the user for input, and correctly update the story line without breaking the formatting.

Saved: 6/15/2025 3:12:42 PM