Submission Worksheet

Submission Data

Course: IT114-450-M2025
Assignment: IT114 Module 3 User Input Challenges
Student: Preet S. (ps55)
Status: Submitted | **Worksheet Progress:** 94%
Potential Grade: 9.80/10.00 (98.00%)
Received Grade: 0.00/10.00 (0.00%)
Started: 6/22/2025 11:20:48 AM
Updated: 6/22/2025 12:32:57 PM
Grading Link: https://learn.ethereallab.app/assignment/v3/IT114-450-M2025/it114-module-3-user-input-challenges/grading/ps55
View Link: https://learn.ethereallab.app/assignment/v3/IT114-450-M2025/it114-module-3-user-input-challenges/view/ps55

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:

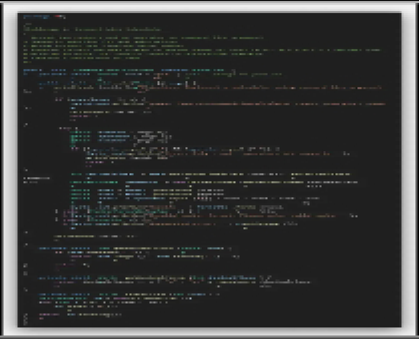
- Don't adjust the give code unless noted
- Challenge 1: Accept two numbers and an operator as command-line arguments (+ and -)
- Challenge 2: Allow integer and floating-point numbers
    - Ensure correct decimal places in output based on input (e.g., 0.1 + 0.2 → 1 decimal place)
- Display an error for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

Part 1:

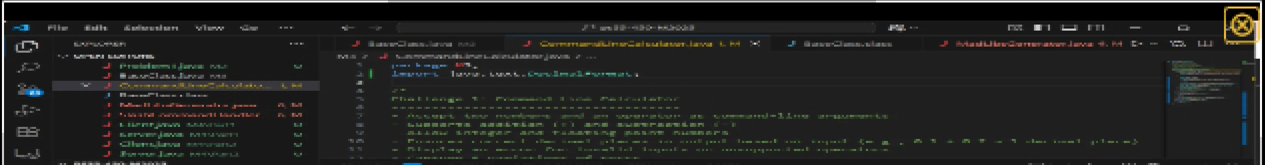
Progress: 100%

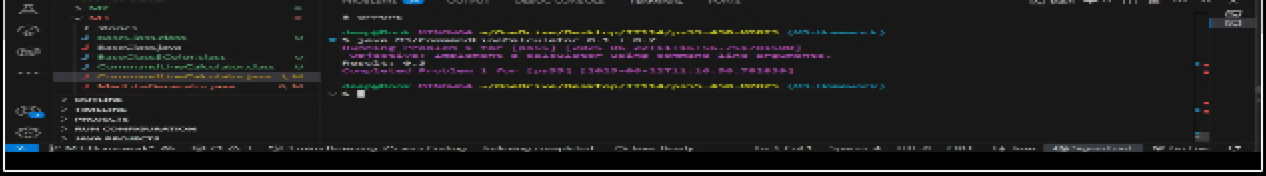
Details:
Two screenshots are expected

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



snippet





output

Saved: 6/22/2025 12:32:06 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/preets4001/ps55-450-M2025/blob/main/M3/CommandLineCalculator.java

URL
https://github.com/preets4001/ps

Saved: 6/22/2025 12:32:06 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 reads two numbers and an operator from the command line, ensuring the user inputs exactly three valid arguments. It calculates the sum or difference based on the operator and determines how many decimal places to display by analyzing the input numbers. Finally, it formats and prints the result with the correct decimal precision while handling any invalid input gracefully

Saved: 6/22/2025 12:32:06 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:

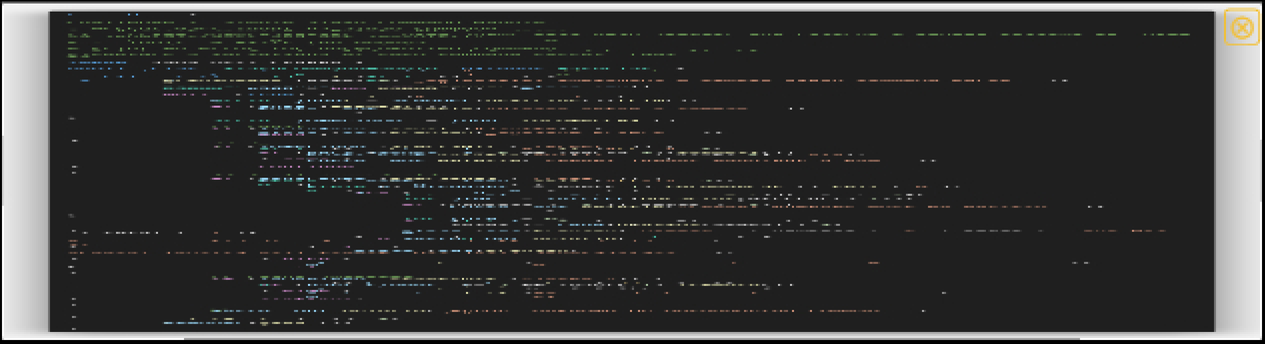
- Don't adjust the give code unless noted
- Challenge 1: Accept user input as slash commands (Commands are case-insensitive)
  - `"/greet <name>" → Prints "Hello, <name>!"`
  - `"/roll <num>d<sides>" → Roll <num> dice with <sides> and returns a`
  - `"/echo <message>" → Prints the message back`
  - `"/quit" → 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%

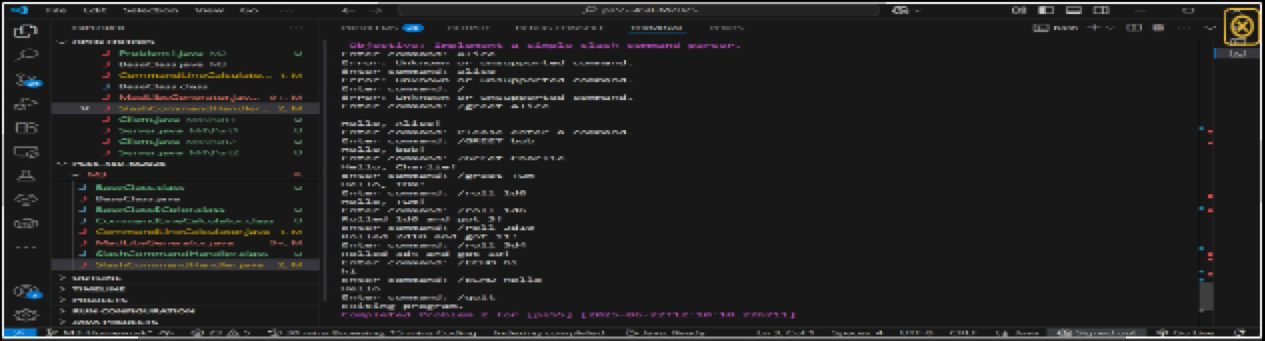
Details:
Two screenshots are expected

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")



```
[illegible]
```

snippet



```
Objective: Implement a simple slash command parser.
Enter command: alice
Error: Unknown or unsupported command.
Enter command: alice
Error: Unknown or unsupported command.
Enter command: /
Error: Unknown or unsupported command.
Enter command: /greet alice

Hello, alice!
Enter command: Please enter a command.
Enter command: /GREET bob
Hello, bob!
Enter command: /greet Charlie
Hello, Charlie!
Enter command: /greet tom
Hello, tom!
Enter command: /roll 1d0
Hello, tom!
Enter command: /roll 1d6
Rolled 1d6 and got 3!
Enter command: /roll 2d20
Rolled 2d20 and got 11!
Enter command: /roll 3d4
Rolled 3d4 and got 10!
Enter command: /echo hi
hi
Enter command: /echo hello
Hello
Enter command: /quit
Exiting program.
Completed Problem 2 for [illegible] [2025-06-22T12:30:30.[illegible]]
```

output

Saved: 6/22/2025 12:32:57 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/preets4001/ps55-450-M2025/blob/main/M3/SlashCommandHandler.java

URL
https://github.com/preets4001/ps

Saved: 6/22/2025 12:32:57 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 breaks the input handling into smaller command patterns by repeatedly checking if the user's input starts with each valid command prefix. For each command, it splits the input to extract arguments, validates the format, and responds appropriately while looping until the /quit command is entered.

Saved: 6/22/2025 12:32:57 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

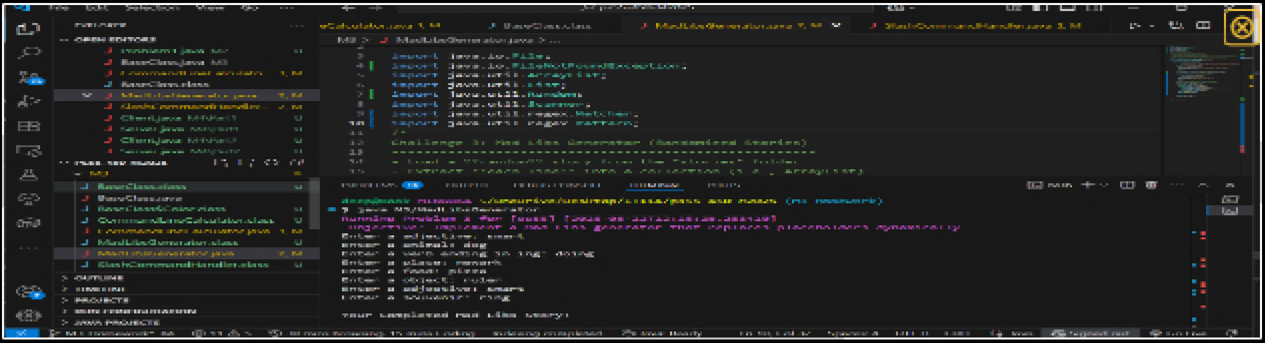
- Any placeholder with underscores should display with spaces instead
- Challenge 4: Replace placeholders with user input (assign back to original slot in collection)
- Add code to solve the problem (add/commit as needed)

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)

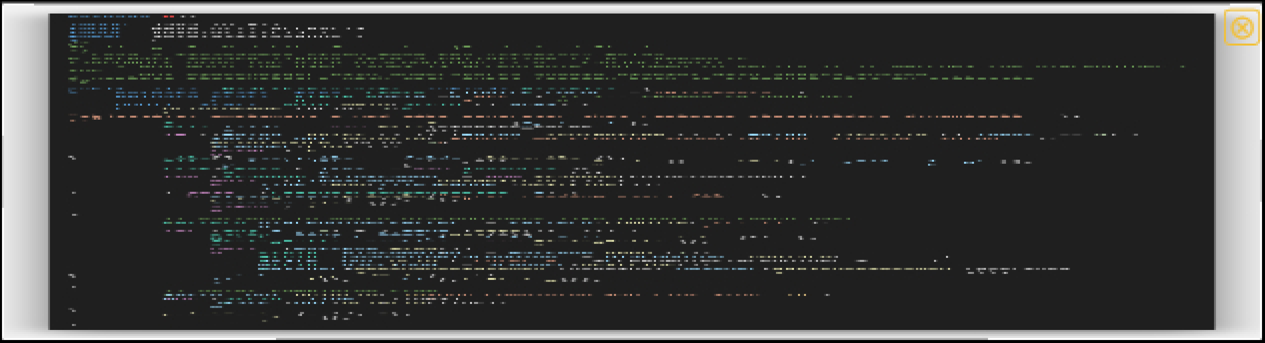


```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.List;
import java.util.Random;
import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
/*
Challenge 3: Mad Libs Generator (Randomized Stories)
[illegible]
```

```
$ java M3/MadLibsGenerator
[illegible]
Enter a adjective: smart
Enter a animal: dog
Enter a verb ending in ing: doing
Enter a place: newark
Enter a food: pizza
Enter a object: ruler
Enter a adjective: smart
Enter a accessory: ring

Your completed mad libs story:
```

output



Snippet

Saved: 6/22/2025 12:31:37 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/preets4001/ps55-450-M2025/blob/main/M3/MadLibsGenerator.java

URL
https://github.com/preets4001/ps

Saved: 6/22/2025 12:31:37 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 first randomly selects a story file from a folder, reads all its lines, and stores them in a list for processing. It uses regular expressions to detect placeholders inside the text and dynamically prompts the user for input to replace each placeholder one by one. By updating the original lines with user-provided values, it builds the completed story while maintaining the structure and flow of the original template

Saved: 6/22/2025 12:31:37 PM

Section #4: (1 pt.) Misc

Progress: 83%

Task #1 (0.33 pts.) - Github Details

Progress: 50%

Part 1:

Progress: 0%

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

Missing Caption

Saved: 6/22/2025 12:31:04 PM

Part 2:

Progress: 100%

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

URL #1
https://github.com/preets4001/ps55-450-M2025/pulls

URL
https://github.com/preets4001/ps

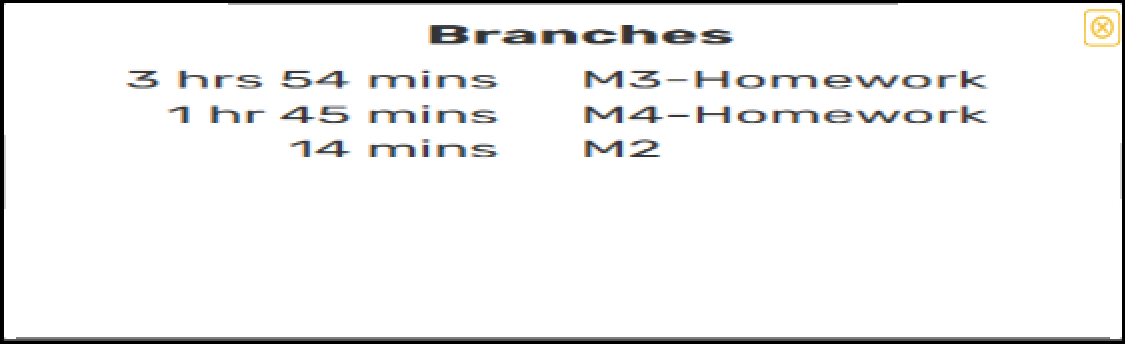
Saved: 6/22/2025 12:31:04 PM

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

Progress: 100%

Details:

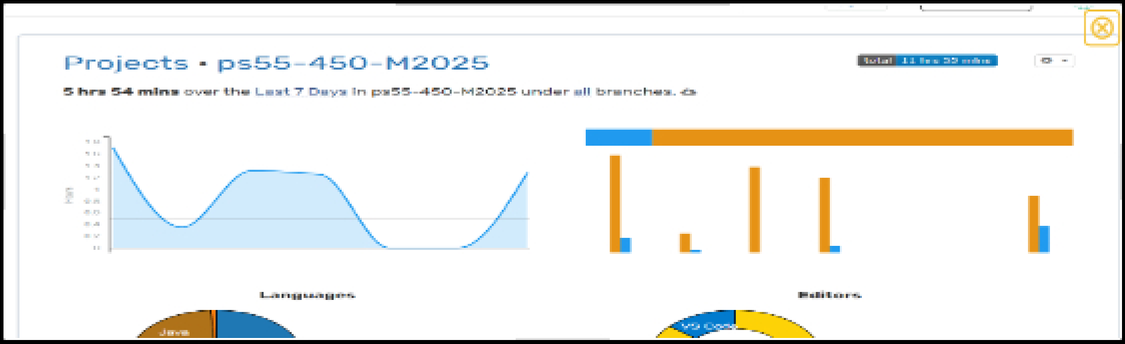
- Visit the WakaTime.com Dashboard
- Click `Projects` and find your repository
- Capture the overall time at the top that includes the repository name
- Capture the individual time at the bottom that includes the file time
- Note: The duration isn't relevant for the grade and the visual graphs aren't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



Branches

3 hrs 54 mins	M3-Homework
1 hr 45 mins	M4-Homework
14 mins	M2

TImespent



Projects • ps55-450-M2025

5 hrs 54 mins over the Last 7 Days in ps55-450-M2025 under all branches.

TimeSpent

Saved: 6/22/2025 12:27:29 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:

I learned how to process user input dynamically and handle different command formats using string manipulation and conditional logic. I also gained experience working with file I/O, collections like ArrayLists, and regular expressions to extract and replace placeholders in text files. Additionally, I practiced importing external classes, managing exceptions, and breaking a problem into smaller, manageable tasks to solve complex challenges more effectively.

Saved: 6/22/2025 12:24:02 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:

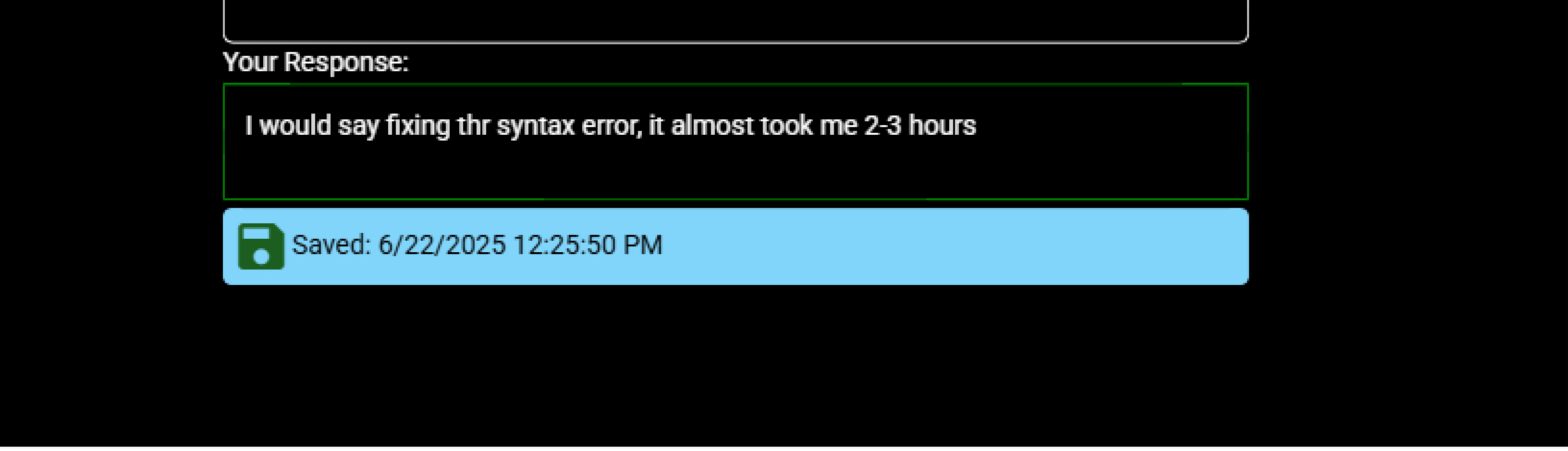
The easiest part of the assignment was handling simple user input and using conditional statements to match the basic commands

Saved: 6/22/2025 12:24:59 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:

I would say fixing thr syntax error, it almost took me 2-3 hours

Saved: 6/22/2025 12:25:50 PM