Submission Worksheet

Submission Data

Course: IT114-450-M2025
Assignment: IT114 Module 3 User Input Challenges
Student: Graham B. (gb373)
Status: Submitted | **Worksheet Progress:** 100%
Potential Grade: 10.00/10.00 (100.00%)
Received Grade: 0.00/10.00 (0.00%)
Started: 6/9/2025 2:38:19 PM
Updated: 6/9/2025 4:06:59 PM
Grading Link: https://learn.ethereallab.app/assignment/v3/IT114-450-M2025/it114-module-3-user-input-challenges/grading/gb373
View Link: https://learn.ethereallab.app/assignment/v3/IT114-450-M2025/it114-module-3-user-input-challenges/view/gb373

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)

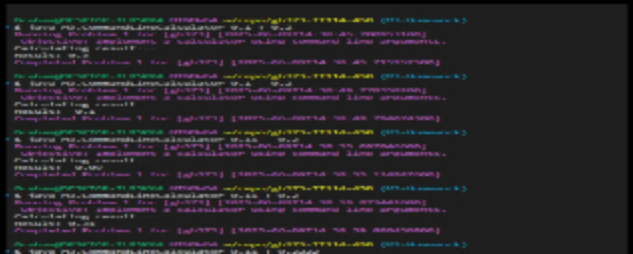
🖼 Part 1:

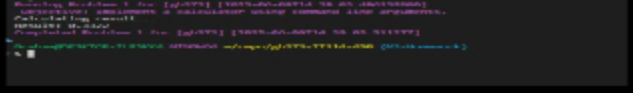
Progress: 100%

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)

Code for CLI





5 Outputs of CLI

Saved: 6/9/2025 2:51:24 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/GrahamBlack10/gb373-IT114-450/blob/M3-Homework/M3/CommandLineCalculator.java

URL
https://github.com/GrahamBlack1

Saved: 6/9/2025 2:51:24 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:

How the code solves the challenge is first variables are defined. Then decimalPlaces finds the length of decimals. Then the if statement makes sure the only operators are + and -. After the next if statement gives the result of the math. Finally, String format makes sure the decimal is the correct length.

Saved: 6/9/2025 2:51:24 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

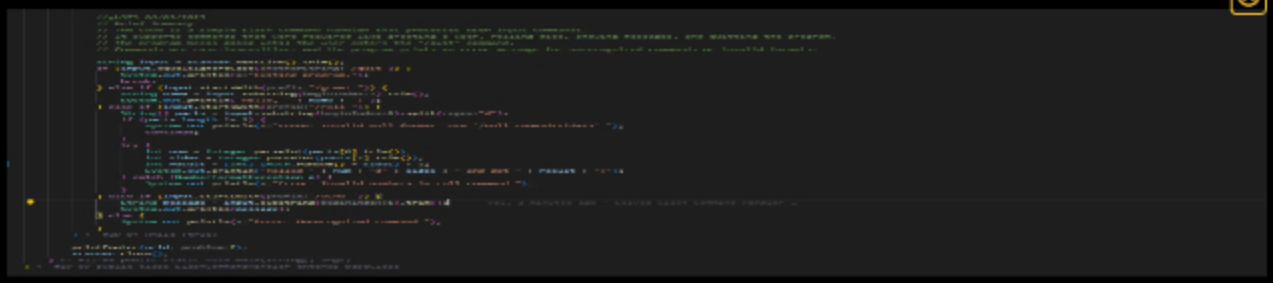
- Challenge 1: Accept user input as slash commands (Commands are case-insensitive)
  - `"/greet <name>" → Prints "Hello, <name>!"`
  - `"/roll <num>d<sides>" → Roll <num> dice with <sides> and returns a`
  - `"/echo <message>" → Prints the message back`
  - `"/quit" → Exits the program`
- Challenge 2: 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:

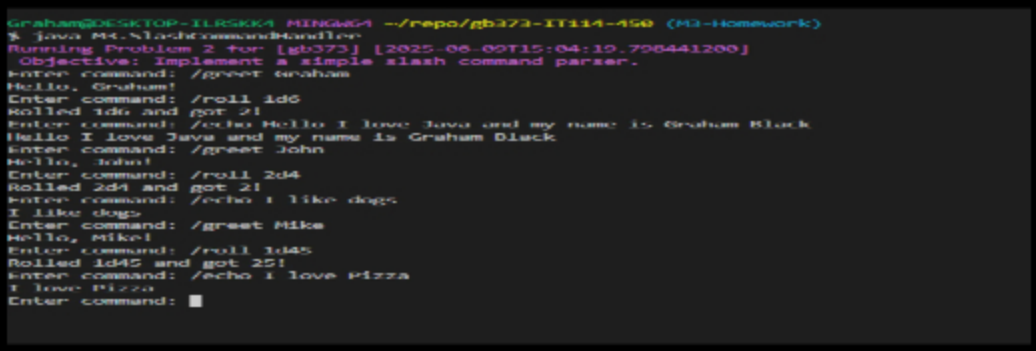
Progress: 100%

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")



Code for Slash



Outputs for Slash

Saved: 6/9/2025 3:15:24 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/GrahamBlack10/gb373-

URL
https://github.com/GrahamBlack1

IT114-450/blob/M3-Homework/M3/SlashCommandHandler.java

Saved: 6/9/2025 3:15:24 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:

How the code solves the challenge is that I use the scanner for the input. If and else statements are used to get the command of the user and do what it's supposed to do while ignoring case. Roll uses a try and catch method for getting a correct random number and making sure the input is correct. The last else statement gives an error for an unrecognizable command.

Saved: 6/9/2025 3:15:24 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)

Code for MadLibs Part 1

Code for MadLibs Part 2

Outputs for MadLibs

Saved: 6/9/2025 3:56:28 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/GrahamBlack10/gb373-IT114-450/blob/M3-Homework/M3/MadLibsGenerator.java

URL
https://github.com/GrahamBlack1

Saved: 6/9/2025 3:56:28 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:

How the code solves the challenge si it first makes sure that there is a story. Then it randomizes what story to give. Then it uses for and while loops to fill in each of the placeholders while making sure it shows the correct one. Finally, it prints the whole story called finalStory with all the placeholders filled in by the user.

Saved: 6/9/2025 3:56:28 PM

Section #4: (1 pt.) Misc

Progress: 100%

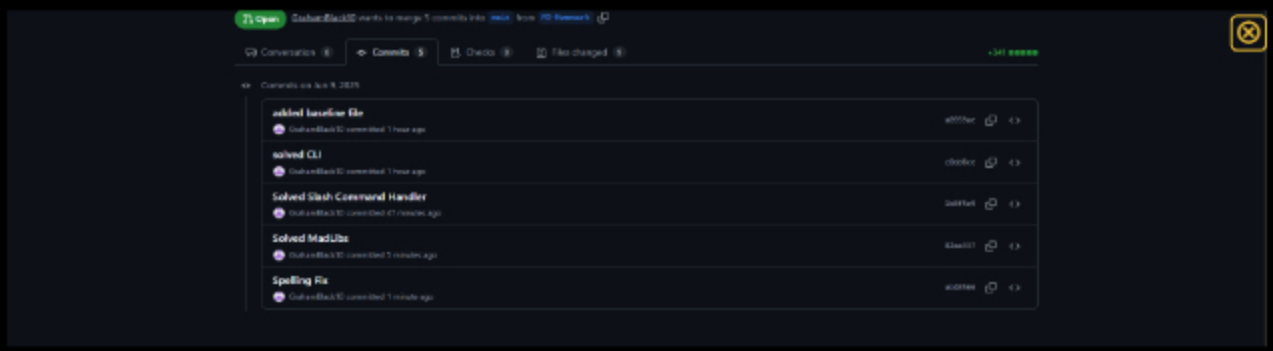
Task #1 (0.33 pts.) - Github Details

Progress: 100%

Part 1:

Progress: 100%

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



Commit History

Saved: 6/9/2025 3:59:31 PM

Part 2:

Progress: 100%

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

URL #1
https://github.com/GrahamBlack10/gb373-IT114-450/pull/4

URL
https://github.com/GrahamBlack1

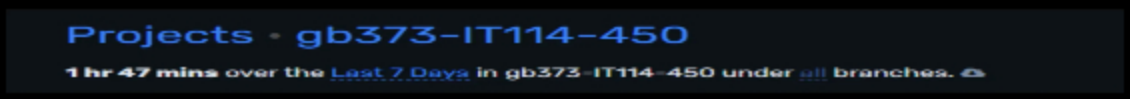
Saved: 6/9/2025 3:59:31 PM

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



Overall Time for Waka

Individual time for Waka

Saved: 6/9/2025 4:02:03 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)

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

Your Response:

I learned more about advanced things in Java. I knew most of the stuff we learned this week, but it was a good refresher. I wanted to learn about scanners, so that was nice to learn about. I also learned about the story of file grabbing from Java. I've never done that before, so that was interesting to learn and do the code for it.

Saved: 6/9/2025 4:04:19 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:

Most to all of this assignment was easy. I found myself knowing what to do and where to put things. I added a lot that helped my assignment work, and I needed that refresher on basic Java from last week.

Saved: 6/9/2025 4:05:32 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 didn't find anything that hard in this assignment. I ran into some small issues here and there when working on each challenge. One of the main ones, and that stuck out, was getting the dice roll command to work properly and give the right output, and accept the right input.

Saved: 6/9/2025 4:06:59 PM