Course: IT114-002-S2025
Assignment: IT114 Module 3 User Input Challenges
Student: Hitarth P. (hp627)
Status: Submitted | Worksheet Progress: 96.30%
Potential Grade: 9.00/10.00 (90.00%)
Received Grade: 0.00/10.00 (0.00%)
Grading Link: https://learn.ethereallab.app/assignment/v3/IT114-002-S2025/it114-module-3-user-input-challenges/grading/hp627

Instructions

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)

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

Combo Task:

Weight: *100%*
Objective: *Edit the `main` method to solve the requirements*

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)

Item:#1

Weight: *40%*

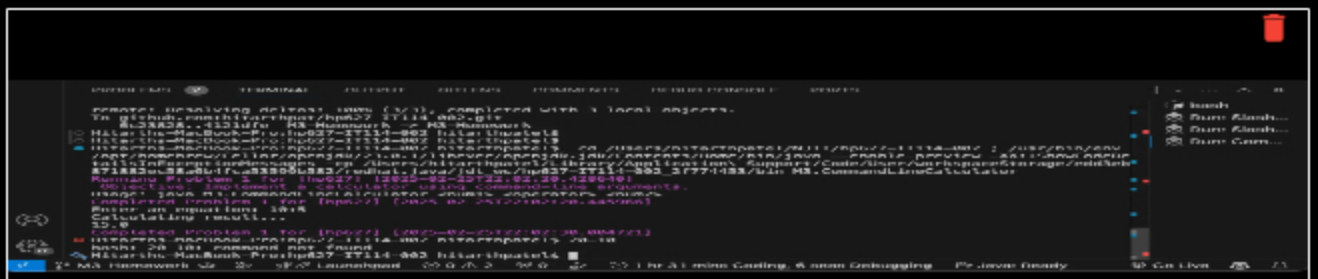
Details:
Two screenshots are expected

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

Image Prompt

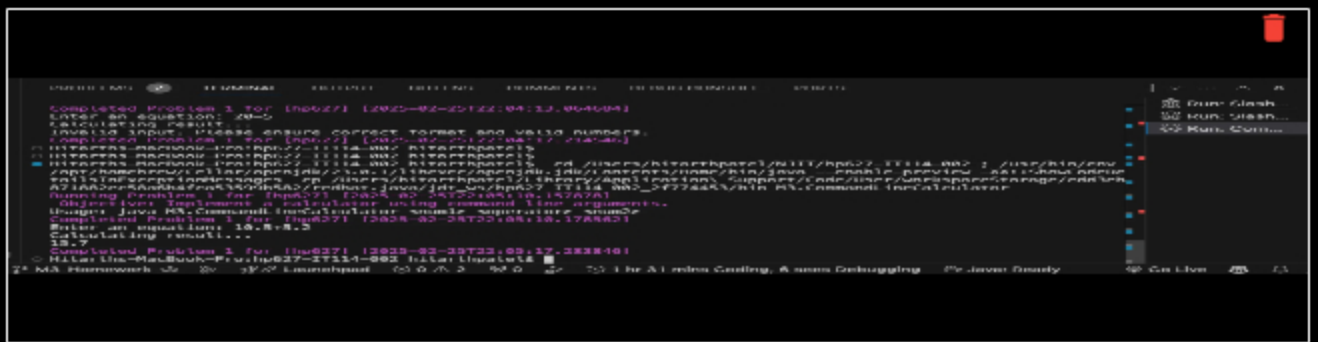
Code

Code

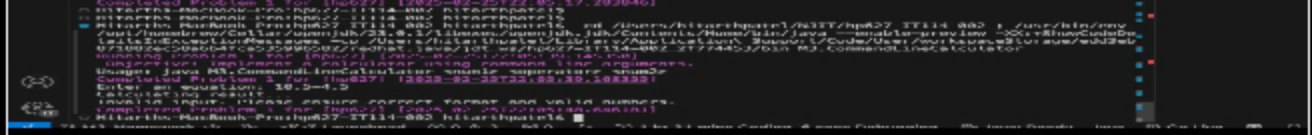


Addition

Subtraction



Floating-Point Addition



Floating-Point Subtraction

Saved: 2/25/2025 10:15:43 PM

Item:#2

Weight: *20%*

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

Url Prompt

URL #1
https://github.com/hitarthpat/hp627-IT114-002/main/M3/CommandLineCalculator.java

URL
https://github.com/hitarthpat/hp62

Saved: 2/25/2025 10:15:43 PM

Item:#3

Weight: *40%*

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

Text Prompt

Your Response:

The code requires the user to enter an equation from addition to subtraction. It scans through the input created by the user and separates in and then the operation and then the second number. If the user entered the wrong format then it will display an error message.

Saved: 2/25/2025 10:15:43 PM

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

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

Combo Task:

Weight: *100%*
Objective: *Edit the `main` method to solve the requirements*

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 single out`
  - `"/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)

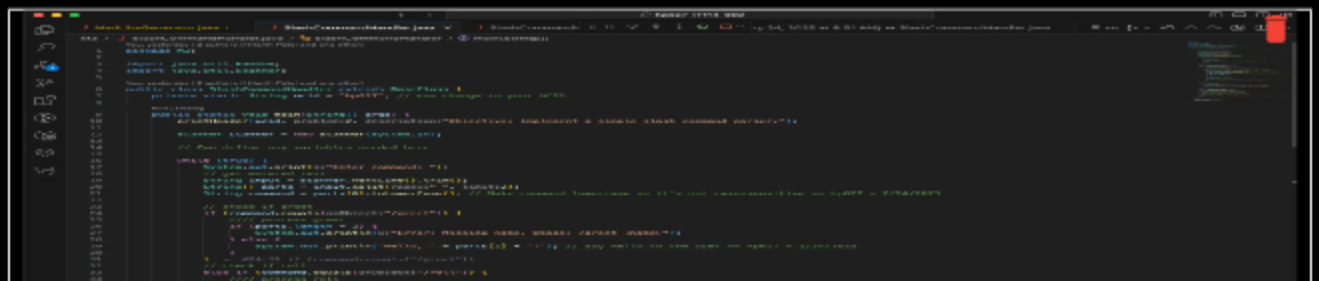
Item:#1

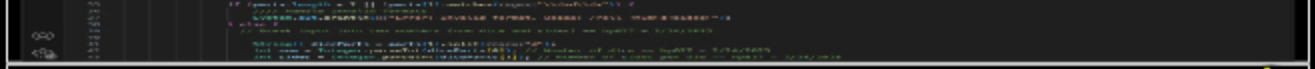
Weight: *40%*

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")

Image Prompt





Code

Code

/greet /roll / echo /quit

Saved: 2/25/2025 10:20:38 PM

Item:#2

Weight: *20%*

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

Url Prompt

URL #1
https://github.com/hitarthpat/hp627-IT114-002/main/M3/SlashCommandHandler.java

URL
https://github.com/hitarthpat/hp62

Saved: 2/25/2025 10:20:38 PM

Item:#3

Weight: *40%*

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

Text Prompt

Your Response:

The code tends to continuously take inputs from the user and process it. While also trimming the unnecessary spaces and the case-inventive. Gives out a clear error message when the user has inputed unrecognized command. The user can input up to 4 commands (/greet, /roll, /echo,/quit)

Saved: 2/25/2025 10:20:38 PM

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

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

Combo Task:

Weight: *100%*
Objective: *Edit the `main` method to solve the challenges*

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)

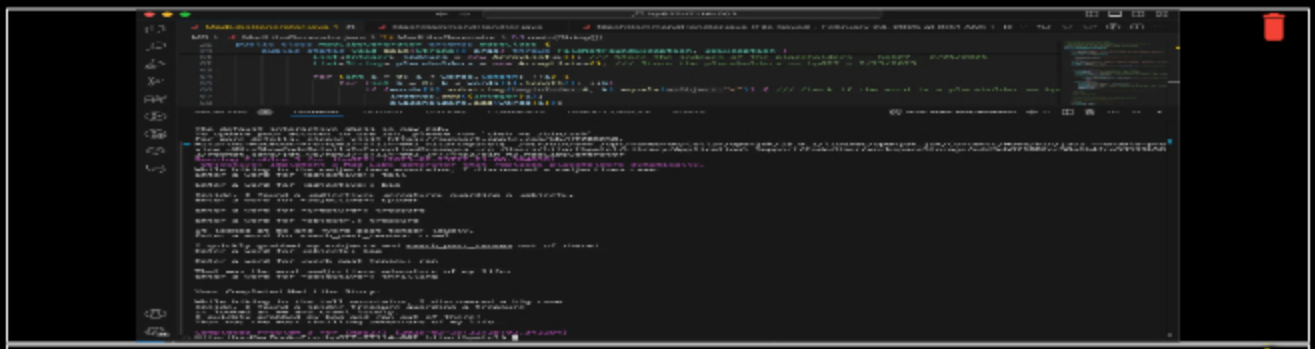
Item:#1

Weight: *40%*

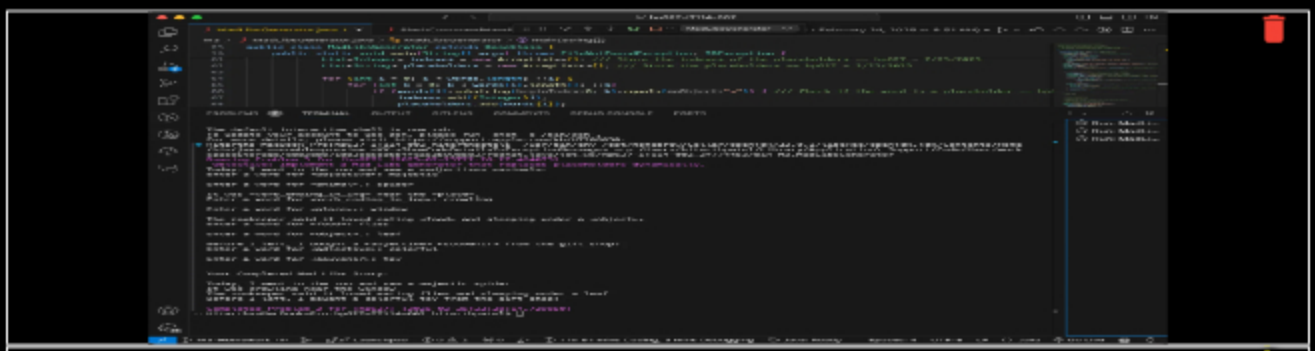
Details:
Two screenshots are expected

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)

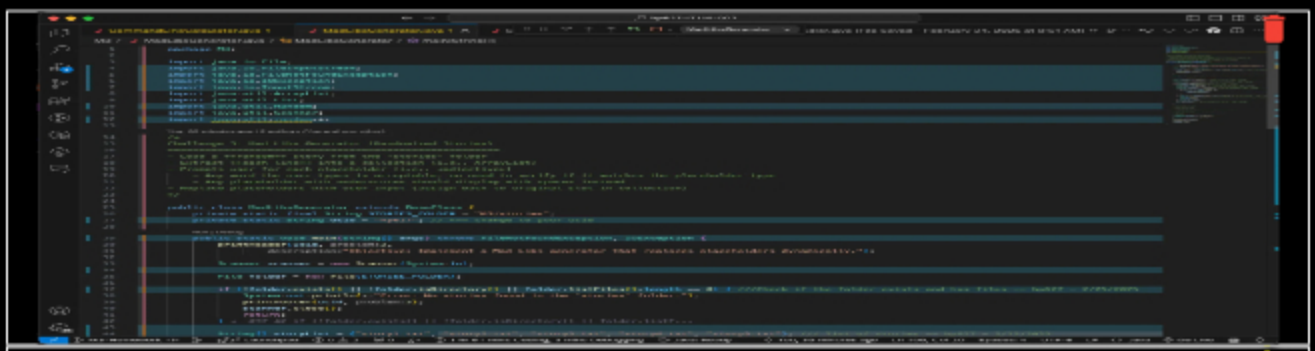
Image Prompt



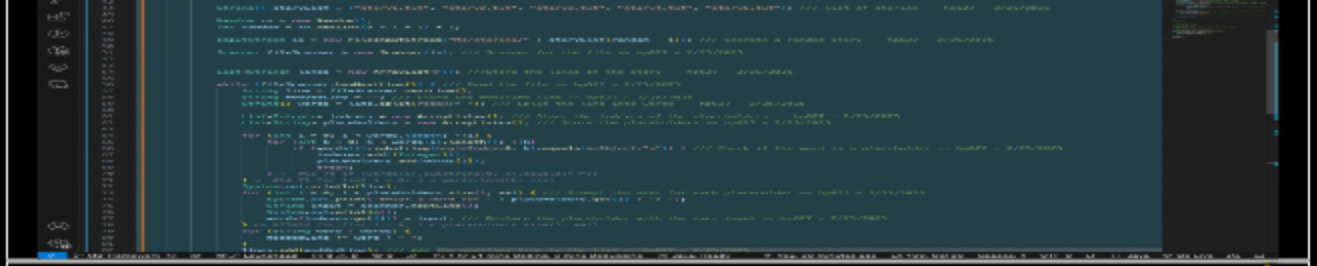
Story



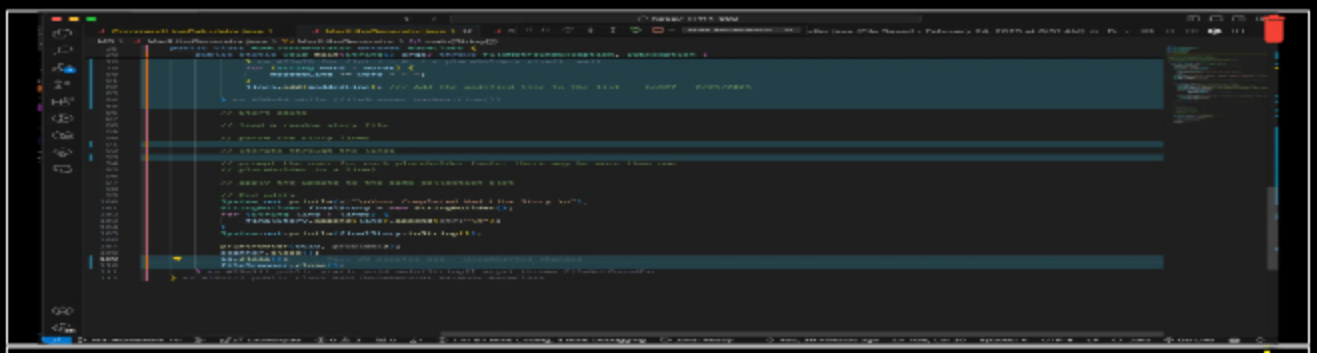
Story 2



Code



Code



Code

Saved: 2/25/2025 10:37:10 PM

Item:#2

Weight: *20%*

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

Url Prompt

URL #1
Missing URL

URL

Saved: 2/25/2025 10:37:10 PM

Item:#3

Weight: *40%*

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

Text Prompt

Your Response:

The program make sure that the stories folders are in baseline code. Based on the stories that are provided is choose a random one. The chosen story text file is read line by line. Each of the line are stored in an Array list which lets the user modify it. The code scans for detect placeholders (). For each placeholder in the story the user can put his own word. The lines are stored from the user input and reconstructed into a complete story.

Saved: 2/25/2025 10:37:10 PM

Section #4: (1 pt.) Misc

Task #1 (0.33 pts.) - Github Details

Combo Task:

Weight: *33.33%*
Objective: *Github Details*

Item:#1

Weight: *60%*

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

Image Prompt

Pull

Saved: 2/25/2025 10:39:50 PM

Item:#2

Weight: *40%*

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

Url Prompt

URL #1
https://github.com/hitarthpat/hp627-IT114-002/

URL
https://github.com/hitarthpat/hp62

Saved: 2/25/2025 10:39:50 PM

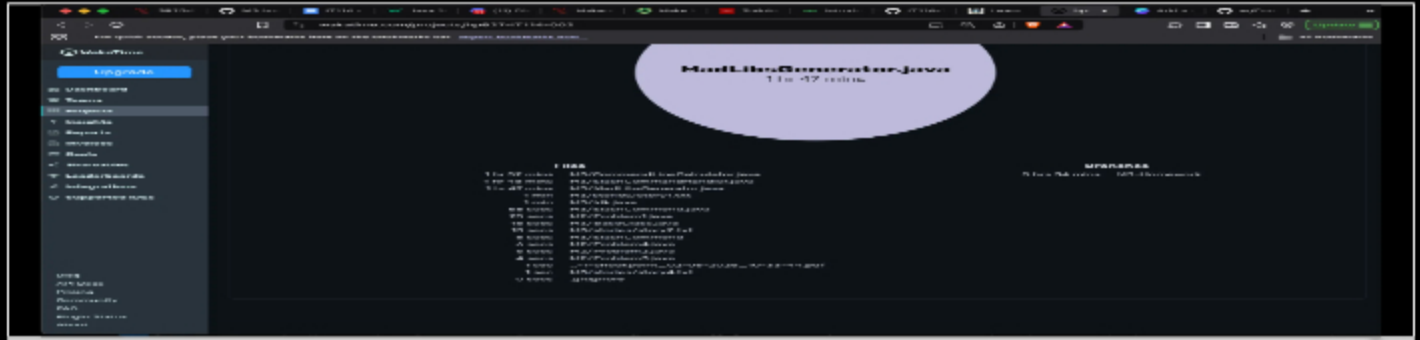
Task #2 (0.00 / 0.33 pts.) - WakaTime - Activity

Weight: *33.33%*
Objective: *WakaTime - Activity*

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

Image Prompt



Waka Time

Saved: 2/25/2025 10:40:35 PM

Task #3 (0.00 / 0.33 pts.) - Reflection

Sub-Tasks:

Task #1 (0.00 / 0.33 pts.) - What did you learn?

Weight: *33.33%*
Objective: *What did you learn?*

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

Text Prompt

Your Response:

I learned check for an existence folder that contained files before using them in a code. Using array to get random selections of story. Using Arraylist to story the story line and modifying it without changing the original story.

Saved: 2/25/2025 10:43:39 PM

Task #2 (0.00 / 0.33 pts.) - What was the easiest part of the a

Weight: *33.33%*
Objective: *What was the easiest part of the assignment?*

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

Text Prompt

Your Response:

The easiest part of the assignment was reading the story file and storing it. Since java has provided with built in methods for files making it easy and using the scanner and fileinputstream that can read the file line by line and storing it.

Saved: 2/25/2025 10:52:28 PM

Task #3 (0.00 / 0.33 pts.) - What was the hardest part of the a

Weight: *33.33%*
Objective: *What was the hardest part of the assignment?*

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

Text Prompt

Your Response:

The hardest part of the code was finding the placeholders using loop. Where I had to manually scan through each word to check for the determined placeholders () to see if it was present. And the loop was attempted to determine the placeholder by looking for the < in the beginning of the substring.

Saved: 2/25/2025 10:50:44 PM