Course: IT114-010-S2025
Assignment: IT114 Module 3 User Input Challenges
Student: Andy C. (aac97)
Status: Submitted | Worksheet Progress: 100.00%
Potential Grade: 10.00/10.00 (100.00%)
Received Grade: 0.00/10.00 (0.00%)
Grading Link: https://learn.ethereallab.app/assignment/v3/IT114-010-S2025/it114-module-3-user-input-challenges/grading/aac97

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)

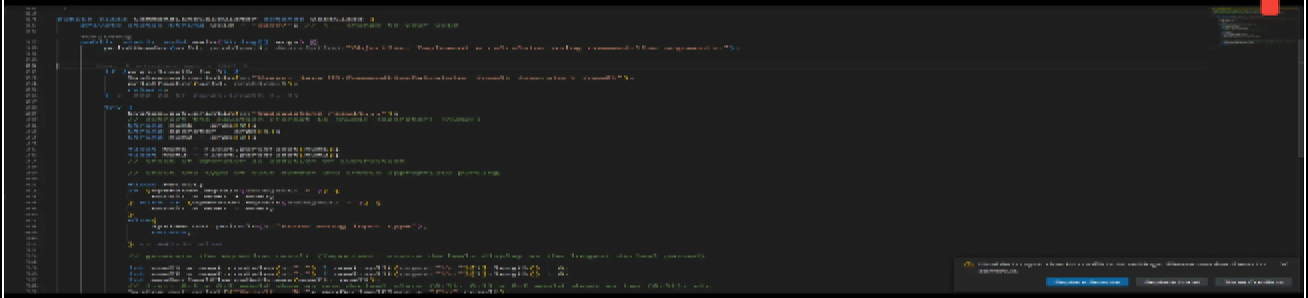
Item:#1

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

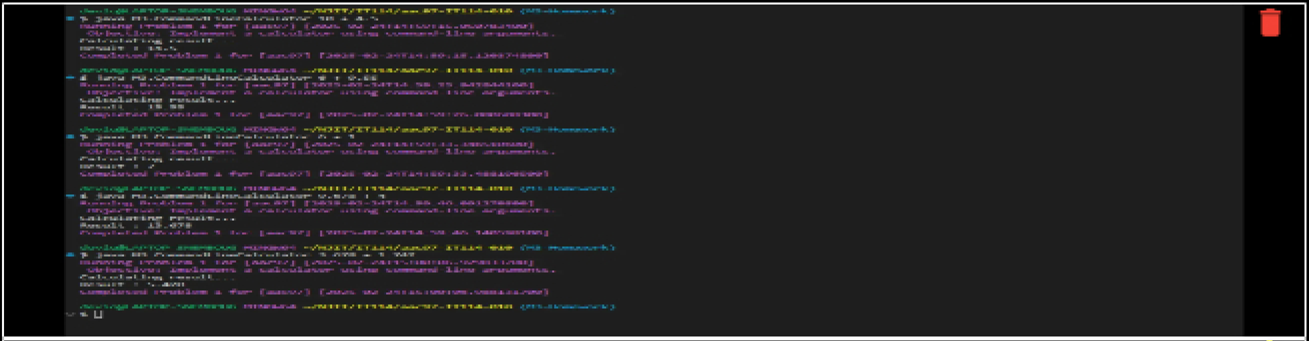


```
[illegible]
```

part 1 code

```
[illegible]
```

part 2 code



```
[illegible]
```

5 examples

Saved: 2/24/2025 3:02:45 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/PolloBear/aac97-IT114-010/M3-Homework/M3/CommandLineCalculator.java

Url
https://github.com/PolloBear/aac9

Saved: 2/24/2025 3:02:45 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:

It collects the information the command line and puts it into differnet object types converting it into a deciaml and adding or subtracting and scanning if there are any decimal places to put it in the right format

Saved: 2/24/2025 3:02:45 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 outcome as "Rolled <num>d<sides> and got <result>!"`
  - `"/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)

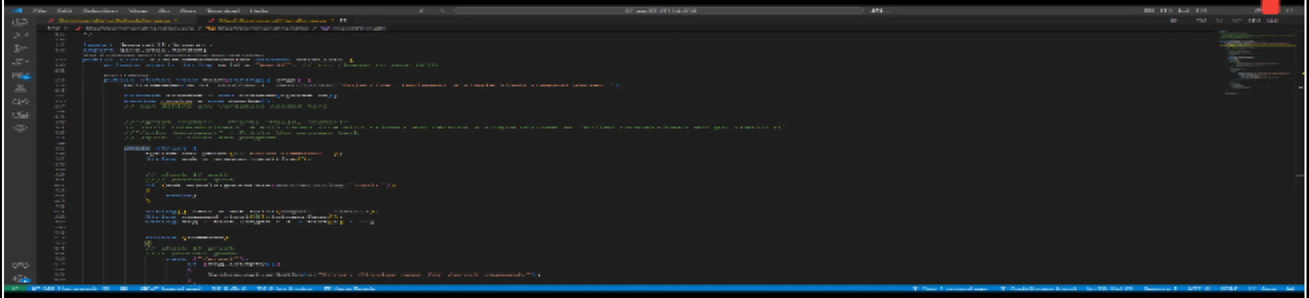
Item:#1

Weight: *40%*

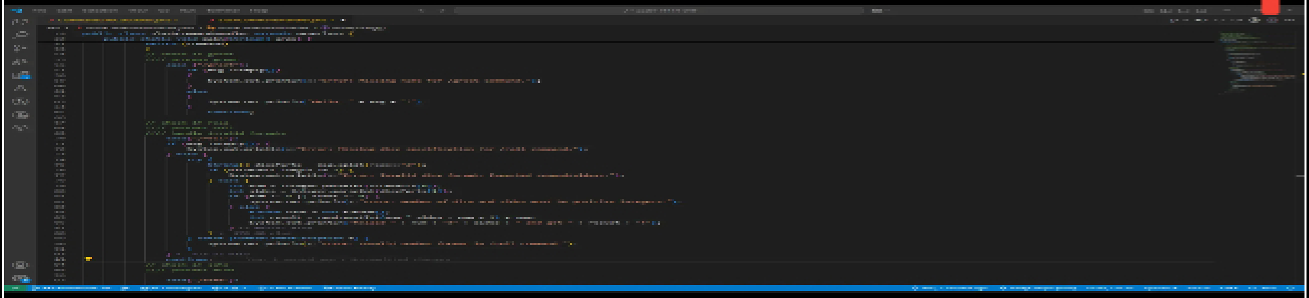
Details:
Two screenshots are expected

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")

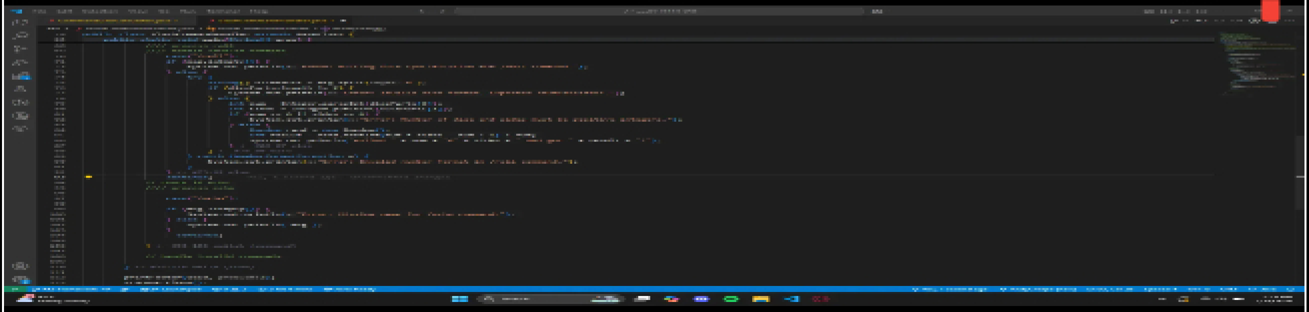
Image Prompt


[illegible]

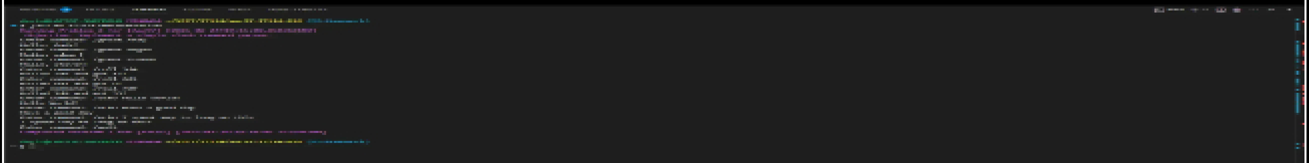
part 1


[illegible]

part 2


[illegible]

part 3


[illegible]

this is the 3 variations of each command

Saved: 2/25/2025 12:30:50 AM

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/PolloBear/aac97-IT114-010M3-Homework/M3/SlashCommandHandler.java

URL
https://github.com/PolloBear/aac9

Saved: 2/25/2025 12:30:50 AM

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 solves the /greet,/roll/ and /echo commands a user puts in when a use provided the write information. If not the user will get an input error telling them they did something wrong. This is put into a switch method so tat it can switch from each case example woult be /greet is one case and the rest being the same

Saved: 2/25/2025 12:30:50 AM

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)

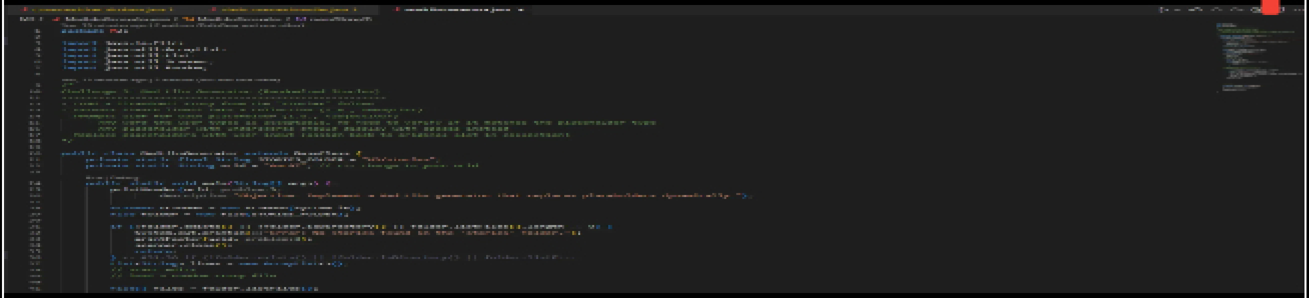
Item:#1

Weight: *40%*

Details:
Two screenshots are expected

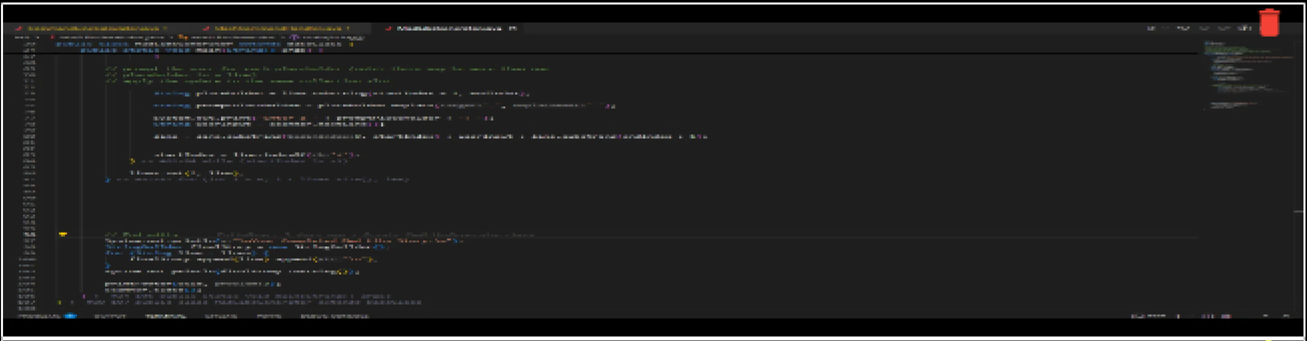
1. Snippet of relevant code showing solution (with ucid/date comment)
2. Full output of executing the program (Capture the process for at least 2 stories)

Image Prompt

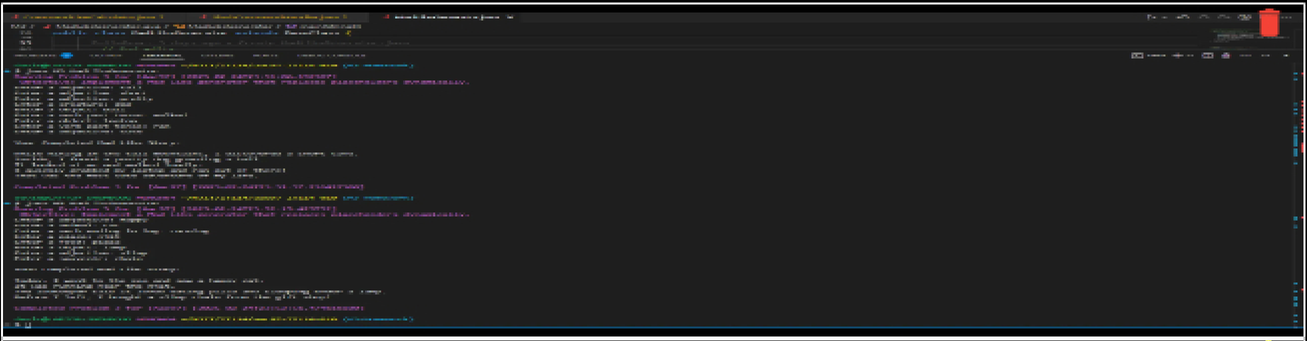

[illegible]

part 1 code

part 2 code



part 3 code



Testing of the code

Saved: 2/25/2025 12:29:44 AM

Item:#2

Weight: *20%*

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

Url Prompt

Url Prompt

URL #1
https://github.com/PolloBear/aac97-IT114-H010M3-Homework/M3/MadLibsGenerator.java

URL
https://github.com/PolloBear/aac9

Saved: 2/25/2025 12:29:44 AM

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 gets a random storie from the files provided which five stories and with that it find all the types in <> and ask the user to put a certain word to fill it in/

Saved: 2/25/2025 12:29:44 AM

Section #4: (1 pt.) Misc

Task #1 (0.33 pts.) - Github Details

Combo Task:

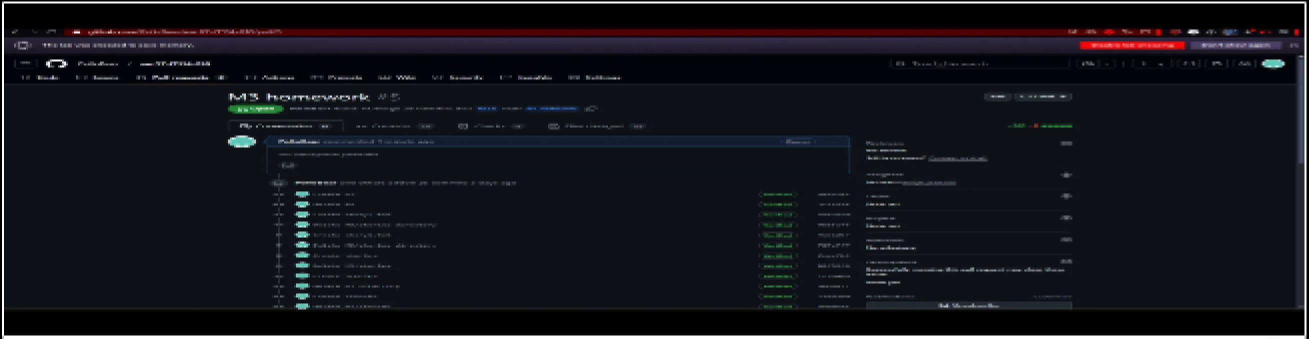
Weight: *33.33%*
Objective: *Github Details*

Item:#1

Weight: *60%*

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

Image Prompt


M3 homework #5

git screenshot

Saved: 2/24/2025 11:55:29 PM

Item:#2

Weight: *40%*

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

Url Prompt

URL #1
[https://github.com/PolloBear/aac97-IT11[illegible]](https://github.com/PolloBear/aac97-IT11)

URL
https://github.com/PolloBear/aac9

Saved: 2/24/2025 11:55:29 PM

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

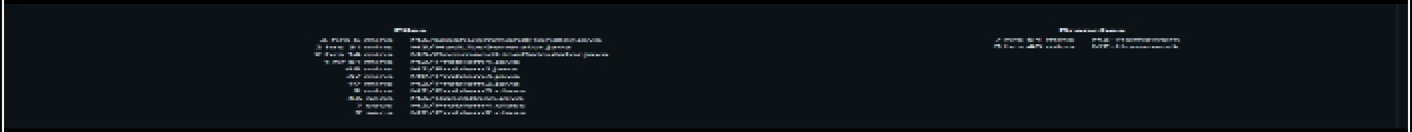
Weight: *33.33%*
Objective: *WakaTime - Activity*

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 12:27:16 AM

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:

Array manipulation and using the files. I had to search things up and figure out how to work with the files so much that I feel like I still don't get it a bit but with more practice I can do it. Using an array to its full potential was honestly fun to figure out i feel like I understood that more than anything and using other commands to grab like the switches.

Saved: 2/25/2025 12:20:11 AM

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:

This wasn't really easy, honestly, but the easiest part was the first one: creating the /greet and /echo, which is a simple print statement, and figuring out if the user put the correct content. That and the first one were the easiest parts compared to the others.

Saved: 2/25/2025 12:20:44 AM

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

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 last one is trying to figure out if there is a more simple way to put the code. Getting the idea to fix it is so much easier than figuring out how to put it into code. I feel like I can find the answer by figuring out how to do the problem but putting it into code is so hard. The files were also hard because forgot how to do it.

Saved: 2/25/2025 12:26:16 AM

[redacted]