

Submission Worksheet

Submission Data

Course: IT114-450-M2025

Assignment: IT114 Module 3 User Input Challenges

Student: Matt T. (mt85)

Status: Submitted | **Worksheet Progress:** 100%

Potential Grade: 10.00/10.00 (100.00%)

Received Grade: 0.00/10.00 (0.00%)

Started: 6/8/2025 10:47:32 PM

Updated: 6/8/2025 11:00:27 PM

Grading Link: <https://localhost:8080/assignment/v3/IT114-450-M2025/it114-module-3-user-input-challenges/grading/mt85>

View Link: <https://localhost:8080/assignment/v3/IT114-450-M2025/it114-module-3-user-input-challenges/view/mt85>

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)

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., $0.1 + 0.2 \rightarrow 1$ 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)

```
26 try {
27     System.out.println(x:"Calculating result...");
28     // extract the equation (format is <num1> <operator> <num2>)
29
30     // check if operator is addition or subtraction
31
32     // check the type of each number and choose appropriate parsing
33
34     // generate the equation result (Important: ensure decimals display as the
35     // longest decimal passed)
36     // i.e., 0.1 + 0.2 would show as one decimal place (0.3), 0.11 + 0.2 would show
37     // as two (0.31), etc
38     // mt85 06-08-2025 You, 1 second ago • Uncommitted changes
39     System.out.println(x:"The answer is pi");
40 } catch (Exception e) {
41     System.out.println(x:"Invalid input. Please ensure correct format and valid numbers.");
42 }
43
44
```

example code

```
Matt@EtherealLab MINGW64 /d/projects/NJIT/2025/summer/IT114-450-M2025 (M3-Homework)
$ java M3.CommandLineCalculator food + food
Running Problem 1 for [mt85] [2025-06-08T22:46:23.869853400]
Objective: Implement a calculator using command-line arguments.
Calculating result...
The answer is pi
Completed Problem 1 for [mt85] [2025-06-08T22:46:23.880874]
```



Saved: 6/8/2025 10:48:18 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/MattToegel/IT114-450-M2025/blob/M3-Homework/M3/CommandLineCalculator.java>



URL

<https://github.com/MattToegel/IT>

Saved: 6/8/2025 10:48:18 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:**

it uses xyz to do abc



Saved: 6/8/2025 10:48:18 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` → Exit 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`)

```

20
21
22 while (true) {
23     // get entered text
24     // M3-06-08T22:50:28.238340500
25     // check if valid
26     // print out result
27
28     // check if null
29     // print out null
30     // handle invalid formats
31
32     // check if valid
33     // print out value
34
35     // check if quit
36     // process quit
37
38     // handle invalid commands
39
40     // delete this condition/block. It's just here so the sample runs without edits
41     if (1==1) {
42         system.out.println("Breaking loop");
43         break;
44     }
45 }
46
47

```

example code

```

Matt@EtherealLab MINGW64 /d/projects/NOIT/2025/summer/IT114-450-M2025 (M3-Homework)
$ java M3.SlashCommandHandler
Running Problem 2 for [mt85] [2025-06-08T22:50:28.238340500]
$ java M3.SlashCommandHandler
Running Problem 2 for [mt85] [2025-06-08T22:50:28.238340500]
Objective: Implement a simple slash command parser.
Enter command: This is an example of unhandled anything
Breaking loop
Completed Problem 2 for [mt85] [2025-06-08T22:50:28.251878900]

Matt@EtherealLab MINGW64 /d/projects/NOIT/2025/summer/IT114-450-M2025 (M3-Homework)
$ java M3.SlashCommandHandler
Running Problem 2 for [mt85] [2025-06-08T22:50:41.420146500]
Objective: Implement a simple slash command parser.
Enter command: This is an example of unhandled anything
Breaking loop
Completed Problem 2 for [mt85] [2025-06-08T22:50:41.436665400]

```

example output

 Saved: 6/8/2025 10:51:58 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/MattToegel/IT114-450-M2025/blob/M3-Homework/M3/SlashCommandHandler.java>



URL

<https://github.com/MattToegel/IT>

 Saved: 6/8/2025 10:51:58 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:

utilized abc to solve each



Saved: 6/8/2025 10:51:58 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)

```
22 // generate random words
23 // generate random words
24 // generate random words
25 // generate random words
26 // generate random words
27 // generate random words
28 // generate random words
29 // generate random words
30 // generate random words
31 // generate random words
32 // generate random words
33 // generate random words
34 // generate random words
35 // generate random words
36 // generate random words
37 // generate random words
38 // generate random words
39 // generate random words
40 // generate random words
41 // generate random words
42 // generate random words
43 // generate random words
44 // generate random words
45 // generate random words
46 // generate random words
47 // generate random words
48 // generate random words
49 // generate random words
50 // generate random words
51 // generate random words
52 // generate random words
53 // generate random words
54 // generate random words
55 // generate random words
56 // generate random words
57 // generate random words
58 // generate random words
59 // generate random words
60 // generate random words
61 // generate random words
62 // generate random words
63 // generate random words
64 // generate random words
65 // generate random words
66 // generate random words
67 // generate random words
68 // generate random words
69 // generate random words
70 // generate random words
71 // generate random words
72 // generate random words
73 // generate random words
74 // generate random words
75 // generate random words
76 // generate random words
77 // generate random words
78 // generate random words
79 // generate random words
80 // generate random words
81 // generate random words
82 // generate random words
83 // generate random words
84 // generate random words
85 // generate random words
86 // generate random words
87 // generate random words
88 // generate random words
89 // generate random words
90 // generate random words
91 // generate random words
92 // generate random words
93 // generate random words
94 // generate random words
95 // generate random words
96 // generate random words
97 // generate random words
98 // generate random words
99 // generate random words
100 // generate random words
```

example solution

```
1 // java MadLibsGenerator
2 // java MadLibsGenerator
3 // java MadLibsGenerator
4 // java MadLibsGenerator
5 // java MadLibsGenerator
6 // java MadLibsGenerator
7 // java MadLibsGenerator
8 // java MadLibsGenerator
9 // java MadLibsGenerator
10 // java MadLibsGenerator
11 // java MadLibsGenerator
12 // java MadLibsGenerator
13 // java MadLibsGenerator
14 // java MadLibsGenerator
15 // java MadLibsGenerator
16 // java MadLibsGenerator
17 // java MadLibsGenerator
18 // java MadLibsGenerator
19 // java MadLibsGenerator
20 // java MadLibsGenerator
21 // java MadLibsGenerator
22 // java MadLibsGenerator
23 // java MadLibsGenerator
24 // java MadLibsGenerator
25 // java MadLibsGenerator
26 // java MadLibsGenerator
27 // java MadLibsGenerator
28 // java MadLibsGenerator
29 // java MadLibsGenerator
30 // java MadLibsGenerator
31 // java MadLibsGenerator
32 // java MadLibsGenerator
33 // java MadLibsGenerator
34 // java MadLibsGenerator
35 // java MadLibsGenerator
36 // java MadLibsGenerator
37 // java MadLibsGenerator
38 // java MadLibsGenerator
39 // java MadLibsGenerator
40 // java MadLibsGenerator
41 // java MadLibsGenerator
42 // java MadLibsGenerator
43 // java MadLibsGenerator
44 // java MadLibsGenerator
45 // java MadLibsGenerator
46 // java MadLibsGenerator
47 // java MadLibsGenerator
48 // java MadLibsGenerator
49 // java MadLibsGenerator
50 // java MadLibsGenerator
51 // java MadLibsGenerator
52 // java MadLibsGenerator
53 // java MadLibsGenerator
54 // java MadLibsGenerator
55 // java MadLibsGenerator
56 // java MadLibsGenerator
57 // java MadLibsGenerator
58 // java MadLibsGenerator
59 // java MadLibsGenerator
60 // java MadLibsGenerator
61 // java MadLibsGenerator
62 // java MadLibsGenerator
63 // java MadLibsGenerator
64 // java MadLibsGenerator
65 // java MadLibsGenerator
66 // java MadLibsGenerator
67 // java MadLibsGenerator
68 // java MadLibsGenerator
69 // java MadLibsGenerator
70 // java MadLibsGenerator
71 // java MadLibsGenerator
72 // java MadLibsGenerator
73 // java MadLibsGenerator
74 // java MadLibsGenerator
75 // java MadLibsGenerator
76 // java MadLibsGenerator
77 // java MadLibsGenerator
78 // java MadLibsGenerator
79 // java MadLibsGenerator
80 // java MadLibsGenerator
81 // java MadLibsGenerator
82 // java MadLibsGenerator
83 // java MadLibsGenerator
84 // java MadLibsGenerator
85 // java MadLibsGenerator
86 // java MadLibsGenerator
87 // java MadLibsGenerator
88 // java MadLibsGenerator
89 // java MadLibsGenerator
90 // java MadLibsGenerator
91 // java MadLibsGenerator
92 // java MadLibsGenerator
93 // java MadLibsGenerator
94 // java MadLibsGenerator
95 // java MadLibsGenerator
96 // java MadLibsGenerator
97 // java MadLibsGenerator
98 // java MadLibsGenerator
99 // java MadLibsGenerator
100 // java MadLibsGenerator
```

example output

 Saved: 6/8/2025 10:57:52 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://localhost/assignment/v3/IT114-450-M2025/package%20M3;%20%20import%20java.io.File%20change%20to%20your%20ucid%20%20%20%20%20%20public%20static%20void%20main\(Strin](https://localhost/assignment/v3/IT114-450-M2025/package%20M3;%20%20import%20java.io.File%20change%20to%20your%20ucid%20%20%20%20%20%20public%20static%20void%20main(Strin)

URL #2


<https://github.com/MattToegel/IT114-450-M2025/blob/M3-Homework/M3/MadLibsGenerator.java>



URL

<https://github.com/MattToegel/IT>



 Saved: 6/8/2025 10:57:52 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:

I used ... steps to solve each thing

 Saved: 6/8/2025 10:57:52 PM

Section #4: (1 pt.) Misc

Progress: 100%

Task #1 (0.33 pts.) - Github Details

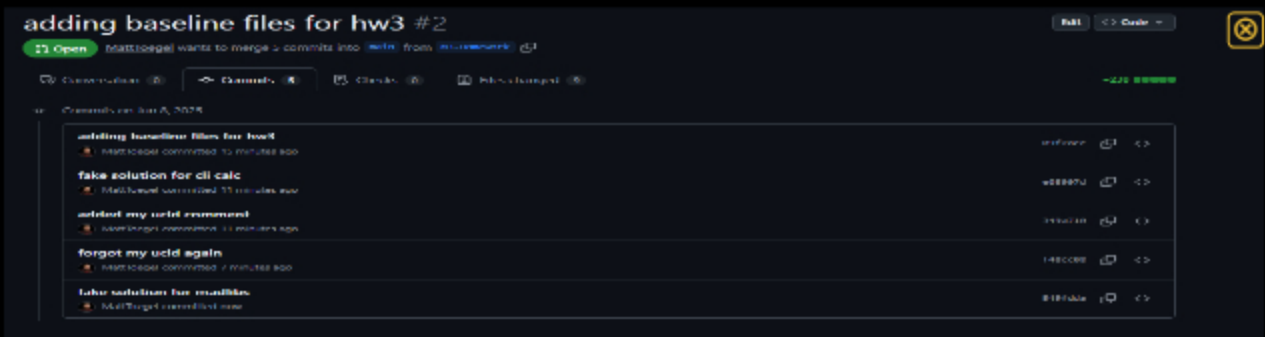
Progress: 100%

Part 1:

Progress: 100%

Details:

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



commit history for hw3

Saved: 6/8/2025 10:58:55 PM

Part 2:

Progress: 100%

Details:

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

URL #1

<https://github.com/MattToegel/IT114-450-M2025/pull/2/commits>



URL

<https://github.com/MattToegel/IT>

Saved: 6/8/2025 10:58:55 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

- Projects • IT114-450-M2025
- 3 hrs 27 mins from Fri Jun 6th until Today in IT114-450-M2025 under all branches. 📄

[illegible]

 Saved: 6/8/2025 11:00:04 PM

Progress: 100%

Progress: 100%

I learned ...

Progress: 100%

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

Your Response:

The easiest part was ...



Saved: 6/8/2025 11:00:19 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 was ..



Saved: 6/8/2025 11:00:27 PM