

Course: IT114-006-S2025

Assignment: IT114 Module 3 User Input Challenges

Student: Muhammad K. (muk)

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/IT114-006-S2025/it114-module-3-user-input-challenges/grading/muk>

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
 1. `git add .`
 2. `git commit -m "adding M3 HW baseline files"`
 3. `git push origin M3-Homework`
 4. 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
 1. Ensure there's a comment with your UCID, date, and brief summary of how the problem was solved
 2. Update the `ucid` variable
 3. 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

```
public static void main(String[] args) {  
    //UCID = BUK  
    //1/24/25  
  
    //extracting the equation  
    double num1 = Double.parseDouble(args[0]); // for num 1  
    String operator = args[1]; // for using either + or - (parseDouble used to  
    double num2 = Double.parseDouble(args[2]); // for num 2  
    turn num1 and output right)  
  
    double total = 0; //initiation variable total  
    if (operator.equals("+")) { //check if the operator is +  
        total = num1 + num2; //add num1 and num2  
    }  
    else if (operator.equals("-")) { //check if num1 and num 2  
        total = num1 - num2; //subtract num 1 and num 2  
    }  
    else {  
        System.out.println("Please enter the right format and numbers"); //if the right format  
        //is not entered  
    }  
  
    DecimalFormat df = new DecimalFormat(pattern:"0.00"); //DecimalFormat is used to ensure decimal  
    System.out.println(df.format(total)); //display as the longest decimal passed
```

Snippet of code





Item:#2

Weight: 20%


Details:

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


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


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


⇒ Url Prompt

URL #1  <https://github.com/MuhammadKhan621/muk-IT114-006/blob/M3-Homework/M3/CommandLineCalculator.java>

<https://github.com/MuhammadKhan621/muk-IT114-006/blob/M3-Homework/M3/CommandLineCalculator.java>

 <https://github.com/MuhammadKhateeb>

 <https://github.com/MuhammadKhateeb>

 <https://github.com/MuhammadKhateeb>



Item:#3

Weight: 40%

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

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

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

⇒ **Text Prompt**

Your Response:

Your Response:

The code ensures that three arguments are provided: the first (a number), the second (either + or -), and the third (another number). If the user does not enter the correct format, they will be given a message. Depending on whether the user enters a + or -, the corresponding operation will be executed in the command line when the user types in the correct format. At the end, DecimalFormat is used to ensure the total is displayed with the correct number of decimal places.

The code ensures that three arguments are provided: the first (a number), the second (either + or -), and the third (another number). If the user does not enter the correct format, they will be given a message. Depending on whether the user enters a + or -, the corresponding operation will be executed in the command line when the user types in the correct format. At the end, DecimalFormat is used to ensure the total is displayed with the correct number of decimal places.



Saved: 2/27/2025 12:33:16 AM

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 uid/ date/comment)
2. Full output of executing the program (Capture 3 variations of each command except `"/quit"`)

⇒ Image Prompt

```
Scanner scanner = new Scanner(System.in);
// Can define any variables needed here
int result = 0;
// Rollid = muk
// Date 2/2/25
while (true) {
    System.out.println("\nEnter command: ");
    // get entered text
    String input = scanner.nextLine(); //Reading user input with scanner
    // process greet
    if (input.matches("greet")) { //if input equals /greet
        System.out.println("Hello, " + input + "!"); //print this message
    }
    // check if roll
    else if (input.matches("roll")) { //check if command starts with /roll
        int numDice = Integer.parseInt(input.substring(2)); //number of dice
        int sides = Integer.parseInt(input.substring(4)); //number of sides
    }
}
```

Snipped of Unfinished Code

```

14 // process roll
15 Random rand = new Random();
16 for (int i = 0; i < numDice; i++) {
17     result += rand.nextInt(sides); //roll the die
18 }
19 System.out.println("Rolled " + numDice + "d" + sides + " and got " + result + "!");
20 }
21 // handle invalid format
22 // check if echo
23 // process echo
24 else if (input.matches("/echo")) {
25     System.out.println();
26 }
27 }
28 }
29 }
30 }
31 }
32 }
33 }
34 }
35 }
36 }
37 }
38 }
39 }
40 }
41 }
42 }
43 }
44 }
45 }
46 }
47 }
48 }
49 }
50 }
51 }
52 }
53 }
54 }
55 }
56 }
57 }
58 }
59 }
60 }
61 }
62 }
63 }
64 }
65 }
66 }
67 }
68 }
69 }
70 }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
```

Snipped of unfinished Code

```

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

Snipped of unfinished Code



Saved: 2/27/2025 12:36:14 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/MuhammadKhan621/muk-IT114-006/blob/M3->



URL

<https://github.com/MuhammadKha>



Saved: 2/27/2025 12:36:14 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:

I was not able to fully finish this.



Saved: 2/27/2025 12:36:14 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



Missing Caption



Saved: 2/27/2025 12:36:45 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/MuhammadKhan621/muk-IT114-006/blob/M3-Homework/M3/MadLibsGenerator.java>



URL

<https://github.com/MuhammadKhan621/muk-IT114-006/blob/M3-Homework/M3/MadLibsGenerator.java>



Saved: 2/27/2025 12:36:45 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:

I was not able to finish this



Saved: 2/27/2025 12:36:45 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

adding M3 HW baseline files #2

view | download

Overall Time



Files

2 hrs 34 mins	M2/CommandLineCalculator.java
2 hrs 22 mins	M2/SlashCommandHandler.java
15 mins	.gitignore
2 mins	M2/Problem2.java
2 mins	M2/MadLibsGenerator.java
2 mins	M2/BaseClass.java
26 secs	M2/Problem3.java
26 secs	.git/COMMIT_EDITMSG
17 secs	M2/Problem1.java
12 secs	M2
2 secs	M2/BaseClass.java
1 sec	M2/CommandLineCalculator.class
0 sec	M2/Problem4.java

Individual Time



Saved: 2/27/2025 12:51:36 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:

I learned how to take input from the command line when running the code rather than using a scanner to ask users for input. I learned how difficult coding can be.



Saved: 2/27/2025 12:54:19 AM

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

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:

Only the CommandLineCalculator assignment was all right because it was straightforward for me.



Saved: 2/27/2025 12:56:28 AM

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

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:

I'd say the SlashCommandHandler and MadLibsGenerator were the hardest even though I was not able to try doing the Madlibs assignment in time. I got a little done for the SlashCommand assignment, but I got stuck on some of it and did not finish all.



Saved: 2/27/2025 1:00:28 AM

