

Submission Worksheet

Submission Data

Course: IT114-005-F2025

Assignment: IT114 Module 3 User Input Challenges

Student: Nilkanth D. (nhd5)

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

Potential Grade: 10.00/10.00 (100.00%)

Received Grade: 0.00/10.00 (0.00%)

Started: 10/15/2025 9:14:28 PM

Updated: 10/15/2025 10:09:50 PM

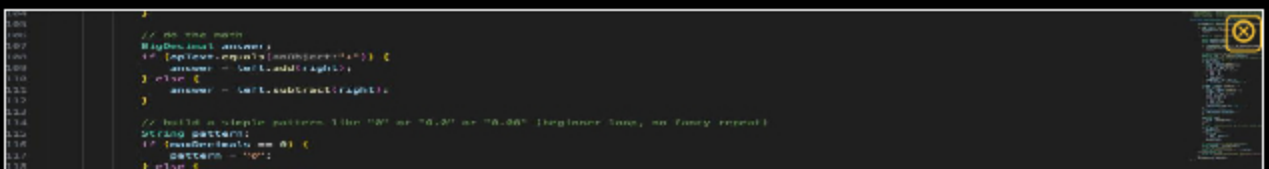
Grading Link: <https://learn.ethereallab.app/assignment/v3/IT114-005-F2025/it114-module-3-user-input-challenges/grading/nhd5>

View Link: <https://learn.ethereallab.app/assignment/v3/IT114-005-F2025/it114-module-3-user-input-challenges/view/nhd5>

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



```

1137         string dots = ".";
1138         int i = 0;
1139         while (i < numvec.size()) {
1140             data[i] = numvec[i] * 100;
1141             i = i + 1;
1142         }
1143         pattern = data;
1144     }
1145
1146     // Format the answer with what's easy to read.
1147     DecimalFormat df = new DecimalFormat(pattern);
1148     String formatted = df.format(answer);
1149
1150     // Watch the screenshot above
1151     System.out.println("The answer is " + formatted);
1152 }

```

code snippet

```

nilkanth@nilkanth-Ds-Macbook-Pro-1564 nhd5-IT114-005 % avac -cp . -d . M3/BaseClass.java M3/CommandLineCalculator.java
java -cp . M3.CommandLineCalculator 0.11 + 0.2
java -cp . M3.CommandLineCalculator 5 - 2.50
zsh: command not found: avac
Running Problem 1 for [nhd5] [2025-10-15T21:18:11.052387]
Objective: Implement a calculator using command-line arguments.
Calculating result...
The answer is 0.31
Completed Problem 1 for [nhd5] [2025-10-15T21:18:11.060909]
Running Problem 1 for [nhd5] [2025-10-15T21:18:11.123938]
Objective: Implement a calculator using command-line arguments.
Calculating result...
The answer is 2.50
Completed Problem 1 for [nhd5] [2025-10-15T21:18:11.130009]

```

code output

```

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

```

code snippet



Saved: 10/15/2025 9:23:53 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/Pixeldepth5/nhd5-IT114-005/M3/M3/CommandLineCalculator.java>



URL

<https://github.com/Pixeldepth5/nhd5-IT114-005/M3/M3/CommandLineCalculator.java>



Saved: 10/15/2025 9:23:53 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)

does)

Your Response:

The program validates the exact CLI format () and uses a try/catch to turn bad inputs into clear errors (W3Schools: methods/main, try/catch). It parses numbers as BigDecimal to avoid floating-point rounding issues and then does either addition or subtraction (BigDecimal basics). To meet the "show longest precision" rule, it counts decimal digits in each input with simple String methods and builds a matching DecimalFormat pattern (Strings, Format Numbers).



Saved: 10/15/2025 9:23:53 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"`)

```
27 public class SlashCommandHandler extends BaseService {
28     private static String ucid = "uc1001" // ucid / uc1001 / uc1002 / uc1003
29 }
30
31 // Challenge 1: Accept user input as slash commands (Commands are case-insensitive)
32 // Challenge 2: Print an error for unrecognized commands
33 // Challenge 3: Print errors for invalid command formats (when applicable)
34 // Add code to solve the problem (add/commit as needed)
35
36 Scanner scanner = new Scanner(System.in);
37
38 while (true) {
39     System.out.print("Enter command: ");
40     String input = scanner.nextLine().trim(); // read line from user
41     // Report longest use as 1
42 }
```

code snippet

code snippet

code snippet

code output

code output



Saved: 10/15/2025 9:43:23 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/Pixeldepth5/nhd5-IT114005M3/M3/SlashCommandHandler.java>



URL

<https://github.com/Pixeldepth5/nhd5-IT114005M3/M3/SlashCommandHandler.java>



Saved: 10/15/2025 9:43:23 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:

The program uses a simple read-evaluate-print loop with Scanner to read each line, splits it into the command and the rest, then handles /greet, /echo, and /roll using basic if/else checks and Math.random() for dice-covering validation, errors, and clean exit. I leaned on W3Schools to structure main and read input (Scanner), to split and parse text with indexOf/substring (Strings), and to generate random numbers (Math), which helped me keep the code beginner-level and compile without extra libraries.



Saved: 10/15/2025 9:43:23 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

code snippet

```
milka@MecBookPro nhd5-IT114-005 % java -cp . M3.MadLibsGenerator
Running Problem 3 for [nhd5] [2025-10-15T21:45:29.376598]
Objective: Implement a Mad Libs generator that replaces placeholders dynamically.
Enter adjective: hot
Enter planet: mars
Enter adjective: cool
Enter verb ending in ing: flying
Enter object: bike
Enter adjective: cold
Enter object: watch
Enter gibberish phrase: gagagaosadovd
Enter verb past tense: presented

Your Completed Mad Libs Story:

I was traveling through space in my hot spaceship when I landed on mars.
The aliens there were cool and flying around a bike.
One of them handed me a cold watch and said, "gagagaosadovd!"
I had no idea what it meant, but I took it and presented back to my spaceship.

Completed Problem 3 for [nhd5] [2025-10-15T21:46:15.606580]
```

code output

```
Running Problem 3 for [nhd5] [2025-10-15T21:54:03.952996]
Objective: Implement a Mad Libs generator that replaces placeholders dynamically.
Enter adjective: cold
Enter adjective: hot
Enter object: car
Enter adjective: tough
Enter verb ending in ing: flying
Enter adjective: rough

Your Completed Mad Libs Story:

A cold witch gave me a potion that would make me hot.
She told me to drink it while standing on a car under the tough noon.
As soon as I drank it, I started flying uncontrollably.
From that day forward, I became the most rough person in town.

Completed Problem 3 for [nhd5] [2025-10-15T21:54:27.272487]
```

code ouput



Saved: 10/15/2025 9:57:38 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/Pixeldepth5/nhd5-IT114-005/M3/M3/MadLibsGenerator.java>



URL

<https://github.com/Pixeldepth5/nhd5-IT114-005/M3/M3/MadLibsGenerator.java>



Saved: 10/15/2025 9:57:38 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:

The program meets the requirements by picking a random story (using Math.random() on either files in M3/stories or a small built-in list), then scanning each line to find , prompting the user with Scanner, and replacing them in place with simple String operations (indexOf, substring, replace).

 Saved: 10/15/2025 9:57:38 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



CS Conversation	→ Console	FS Checks	Files changed	view details
COMMIT HISTORY				
adding M3 NEW baseline files			FILE	1.1
added a solution for M3 CommandLineCalculator.java			FILE	1.1
added a solution for M3 CommandLineCalculator.java			FILE	1.1
Added a solution for MathLibGenerator.java			FILE	1.1
Added a solution for MathLibGenerator.java			FILE	1.1
added a solution for ClockCommandHandler.java			FILE	1.1
Added a solution for ClockCommandHandler.java			FILE	1.1
Added a solution for MathLibGenerator.java			FILE	1.1

commit history

 Saved: 10/15/2025 9:58:45 PM

↪ Part 2:

Progress: 100%

Details:

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


URL #1

<https://github.com/Pixeldepth5/nhd5-IT114@05/commits>



URL

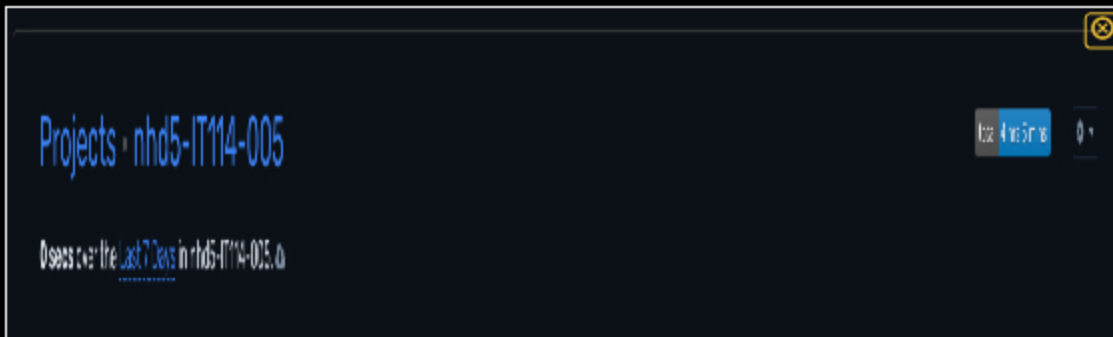
<https://github.com/Pixeldepth5/nhd5-IT114@05/pull/1>

 Saved: 10/15/2025 9:58:45 PM

Task #2 (0.33 pts.) - 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



extension wasnt working , so i reinstalled it



Saved: 10/15/2025 10:02:33 PM

≡ Task #3 (0.33 pts.) - Reflection

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

Details:

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

Your Response:

I learned how to read input from the command line and with Scanner, and how to split a line so I can pick out a command and its arguments. I also figured out how to use BigDecimal and DecimalFormat to keep the right number of decimals, and how to loop through strings to find and replace for the Mad Libs. W3Schools really helped me understand the basics for methods/main, user input, strings, math, and formatting



Saved: 10/15/2025 10:07:43 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:

The easiest part was writing the simple if/else checks for the slash commands and printing messages back to the user. Using Scanner to read a line and basic string methods like trim, indexOf, and substring. Also using examples from W3 schools wiring /greet and /echo was quick.



Saved: 10/15/2025 10:09:10 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 handling numbers and formatting correctly, plus making the Mad Libs pick a random story every run. I had to be careful with decimal places in the calculator and learned that BigDecimal avoids weird rounding, and DecimalFormat prints the exact number of places



Saved: 10/15/2025 10:09:50 PM