Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-006-S2024/it114-number-guesser-4/grade/sb57

IT114-006-S2024 - [IT114] Number Guesser 4

Submissions:

Submission Selection

1 Submission [active] 4/30/2024 12:44:07 AM

Instructions

COLLAPSE

- Create the below branch name
- Implement the NumberGuess4 example from the lesson/slides
  - https://gist.github.com/MattToegel/aced06400c812f13ad030db9518b399f
- Add/commit the files as-is from the lesson material (this is the base template). You may want to push this commit so you can open the pull request and keep it open.
- Pick two (2) of the following options to implement
  - Display higher or lower as a hint after a wrong guess (only after a wrong guess that doesn't roll back the level)
  - Implement anti-data tampering of the save file data (reject user direct edits)
  - Add a difficulty selector that adjusts the max strikes per level (i.e., "easy" 10 strikes, "medium" 5 strikes, "hard" 3 strikes)
  - Display a cold, warm, hot indicator based on how close to the correct value the guess is (example, 10 numbers away is cold, 5 numbers away is warm, 2 numbers away is hot; adjust these per your preference) Only display this when the wrong guess doesn't roll back the level
  - Add a hint command that can be used once per level and only after 2 strikes have been used that reduces the range around the correct number (i.e., number is 5 and range is initially 1-15, new range could be 3-8 as a hint)
  - Implement separate save files based on a "What's your name?" prompt at the start of the game (each person gets their own save file based on user's name)
- Fill in the below deliverables
- Save changes and export PDF
- Git add/commit/push your changes to the HW branch
- Create a pull request to main
- Complete the pull request (don't forget to locally checkout main and pull changes to prep for future work)
- Upload the same PDF to Canvas

Branch name: M3-NumberGuesser-4

Tasks: 7 Points: 10.00

● ^ COLLAPSE ^

Implementation 1 (4 pts.)

● ^ COLLAPSE ^

Task #1 - Points: 1

Text: Chosen Option and Details

Checklist *The checkboxes are for your own tracking

#	Points	Details
□ #1	1	Mention which option you picked
□ #2	1	Explain the logic of how you solved/implemented the chosen option (concrete details). Explain how the code works, don't just paste code snippets

Response:

I picked option 1. Display higher or lower as a hint after a wrong guess (only after a wrong guess that doesn't roll back the level). In order to implement this feature, I edited the processGuess method. I added another if/else statement inside the existing one to compare the input to the number and print whether to go higher or lower with the next guess based off of whether the input was higher or lower than the number.

● ^ COLLAPSE ^

Task #2 - Points: 1

Text: 2+ Screenshots of code and demo

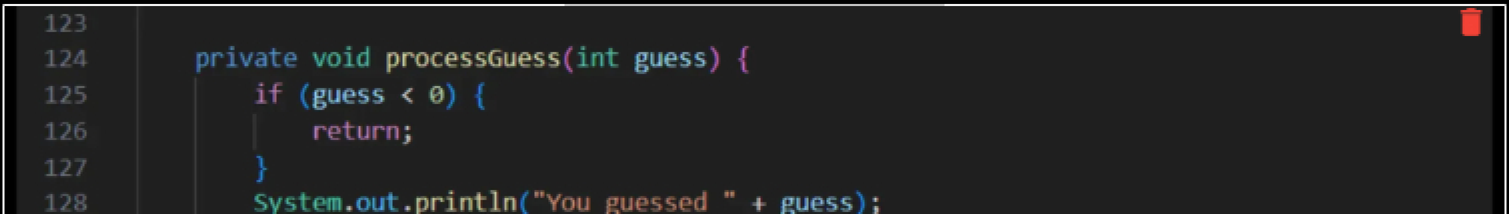
Checklist *The checkboxes are for your own tracking

#	Points	Details
□ #1	1	Show implementation working by running the program
□ #2	1	Clearly caption the screenshot of what you're showing
□ #3	1	The code screenshot(s) clearly show the code specific to the feature
□ #4	1	A comment with the UCID/date is visible near the code change(s)

Task Screenshots:

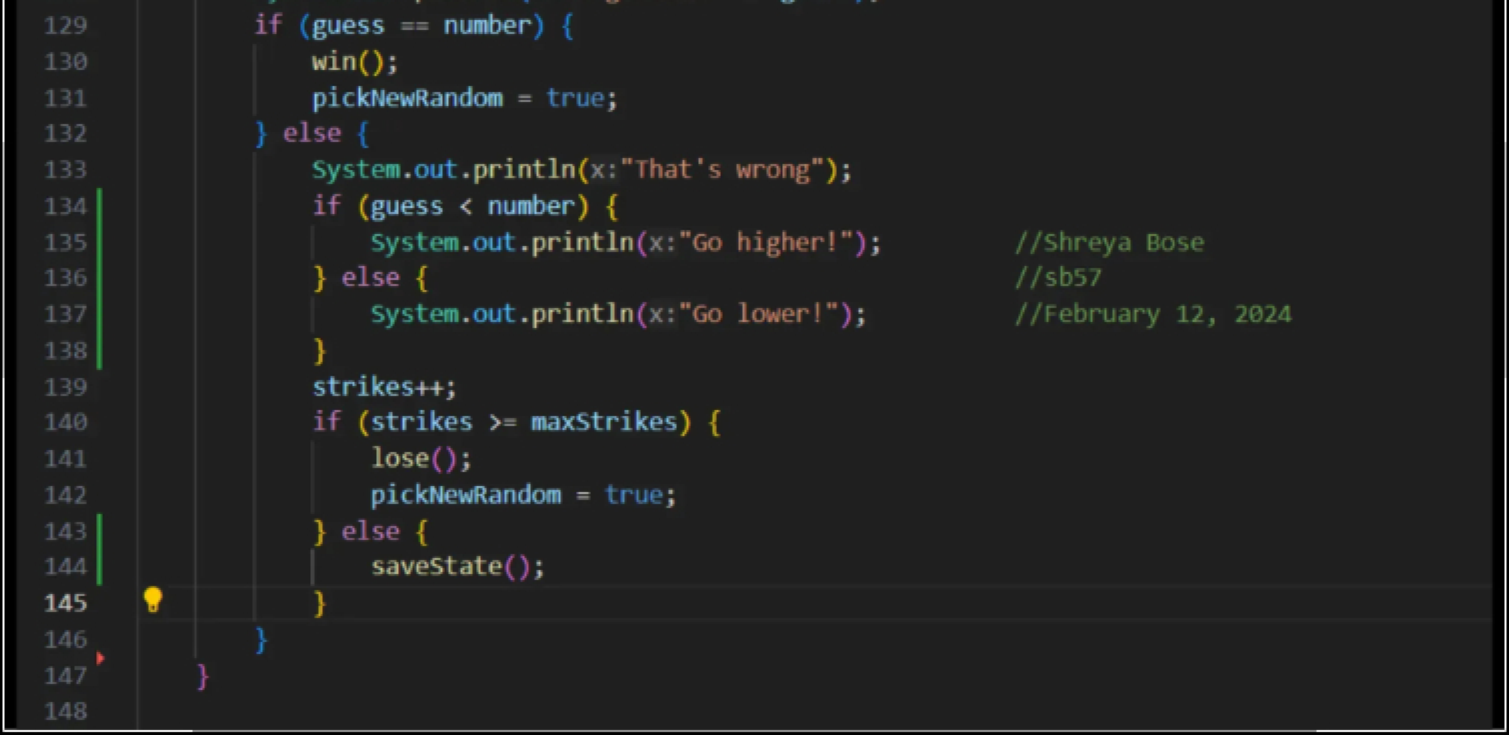
Gallery Style: Large View

Small Medium Large



```

    private void processGuess(int guess) {
        if (guess < 0) {
            return;
        }
        System.out.println("You guessed " + guess);
```



```java
    if (guess == number) {
        win();
        pickNewRandom = true;
    } else {
        System.out.println(x:"That's wrong");
        if (guess < number) {
            System.out.println(x:"Go higher!");         //Shreya Bose
        } else {                                        //sb57
            System.out.println(x:"Go lower!");          //February 12, 2024
        }
        strikes++;
        if (strikes >= maxStrikes) {
            lose();
            pickNewRandom = true;
        } else {
            saveState();
        }
    }
}
```

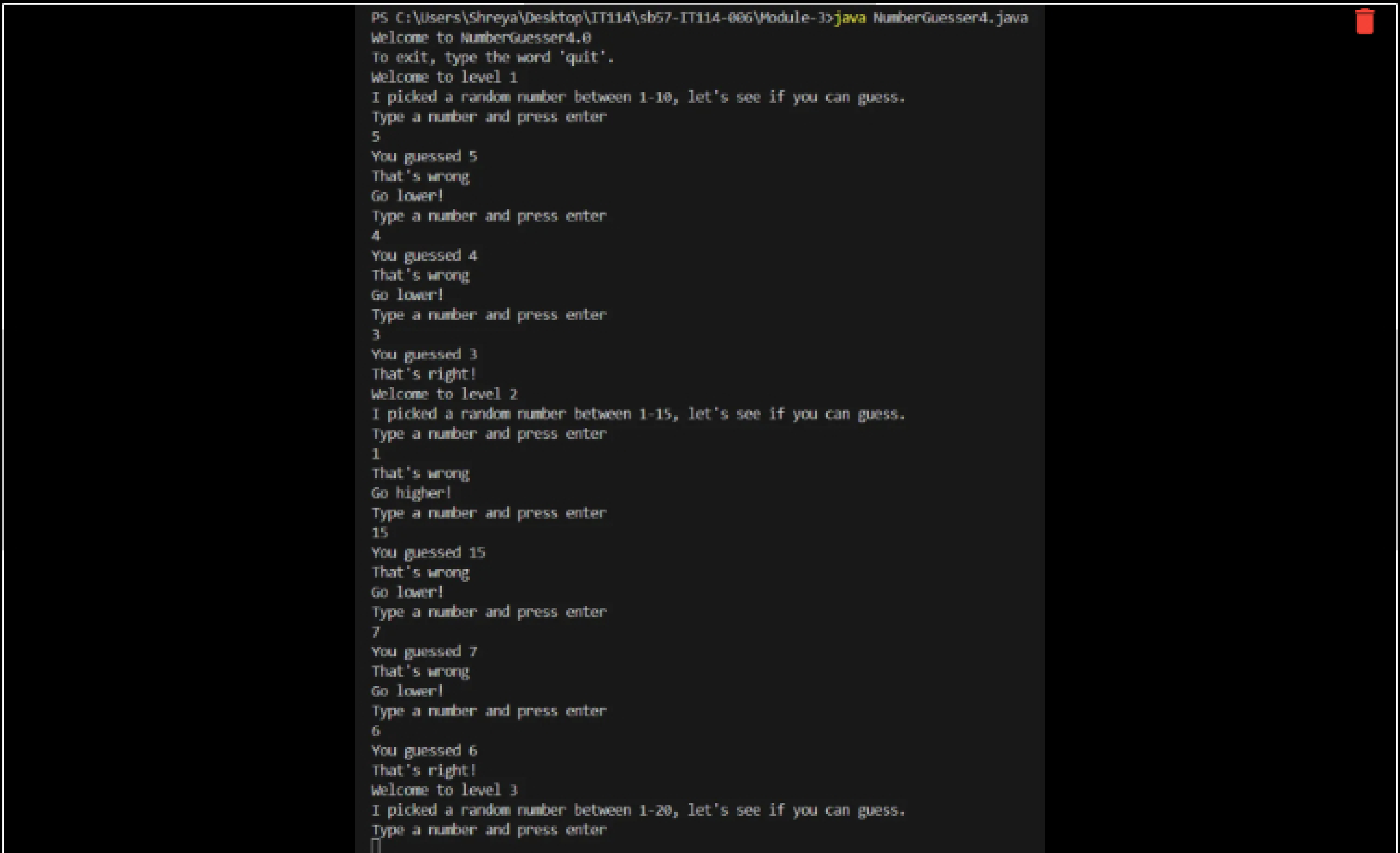
This screenshot specifically shows a snippet of the processGuess method which is what I edited in order to implement a hint for wrong answers based on the guess.

Checklist Items (3)

#2 Clearly caption the screenshot of what you're showing

#3 The code screenshot(s) clearly show the code specific to the feature

#4 A comment with the UCID/date is visible near the code change(s)



```
PS C:\Users\Shreya\Desktop\IT114\sb57-IT114-006\Module-3>java NumberGuesser4.java
Welcome to NumberGuesser4.0
To exit, type the word 'quit'.
Welcome to level 1
I picked a random number between 1-10, let's see if you can guess.
Type a number and press enter
5
You guessed 5
That's wrong
Go lower!
Type a number and press enter
4
You guessed 4
That's wrong
Go lower!
Type a number and press enter
3
You guessed 3
That's right!
Welcome to level 2
I picked a random number between 1-15, let's see if you can guess.
Type a number and press enter
1
That's wrong
Go higher!
Type a number and press enter
15
You guessed 15
That's wrong
Go lower!
Type a number and press enter
7
You guessed 7
That's wrong
Go lower!
Type a number and press enter
6
You guessed 6
That's right!
Welcome to level 3
I picked a random number between 1-20, let's see if you can guess.
Type a number and press enter
```

In this screenshot, I play the game in the terminal after the implementation. In the first round, my guesses were too high so it asked me to go lower until I guessed the right number. In the second round, I first guessed too low, and it asked me to go higher. I then guessed too high so it asked me to go lower.

Checklist Items (2)

#1 Show implementation working by running the program

#2 Clearly caption the screenshot of what you're showing

● ^ COLLAPSE ^

Implementation 2 (4 pts.)

● ^ COLLAPSE ^

Task #1 - Points: 1

Text: Chosen Option and Details

Checklist *The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Mention which option you picked
☐ #2	1	Explain the logic of how you solved/implemented the chosen option (concrete details). Explain how the code works, don't just paste code snippets

Response:

I picked option 4. Display a cold, warm, hot indicator based on how close to the correct value the guess is (example, 10 numbers away is cold, 5 numbers away is warm, 2 numbers away is hot; adjust these per your preference) Only display this when the wrong guess doesn't roll back the level. In order to implement this feature, I edited the processGuess method again. I first find the absolute value of the difference between the number and the incorrect guess. I then once again added another if/else if/else statement inside the existing one. If the difference is greater than 5, it prints "Cold", if it is greater than 2 but less than or equal to 5, it prints "Warm", otherwise it prints "Hot".

● ^ COLLAPSE ^

Task #2 - Points: 1

Text: 2+ Screenshots of code and demo

Checklist *The checkboxes are for your own tracking

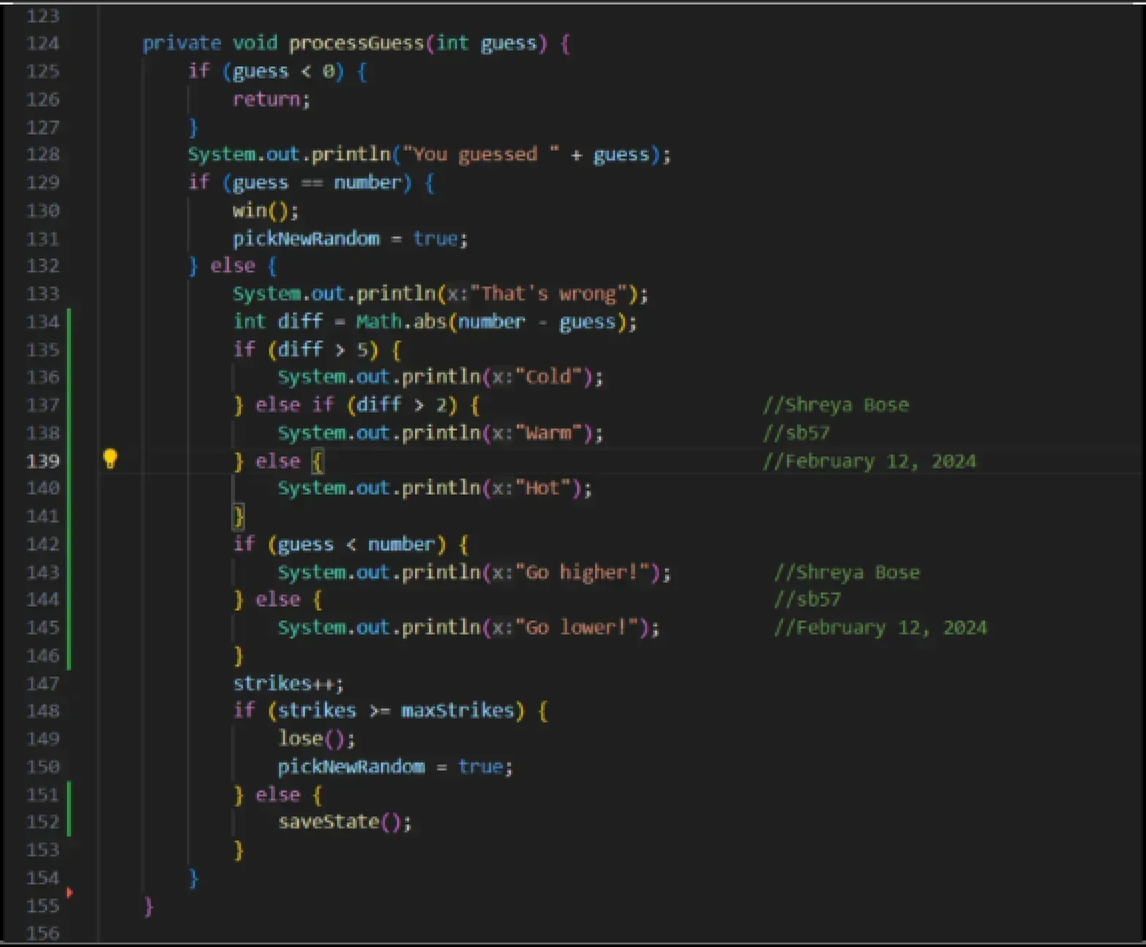
#	Points	Details
☐ #1	1	Show implementation working by running the program
☐ #2	1	Clearly caption the screenshot of what you're showing
☐	1	The code screenshot(s) clearly show the code specific to the feature

#3	1	The code screenshot(s) clearly show the code specific to the feature
🟦 #4	1	A comment with the UCID/date is visible near the code change(s)

Task Screenshots:

Gallery Style: Large View

Small Medium Large



```
    private void processGuess(int guess) {
        if (guess < 0) {
            return;
        }
        System.out.println("You guessed " + guess);
        if (guess == number) {
            win();
            pickNewRandom = true;
        } else {
            System.out.println(x:"That's wrong");
            int diff = Math.abs(number - guess);
            if (diff > 5) {
                System.out.println(x:"Cold");
            } else if (diff > 2) {                          //Shreya Bose
                System.out.println(x:"Warm");               //sb57
            } else {                                        //February 12, 2024
                System.out.println(x:"Hot");
            }
            if (guess < number) {
                System.out.println(x:"Go higher!");          //Shreya Bose
            } else {                                         //sb57
                System.out.println(x:"Go lower!");           //February 12, 2024
            }
            strikes++;
            if (strikes >= maxStrikes) {
                lose();
                pickNewRandom = true;
            } else {
                saveState();
            }
        }
    }
```

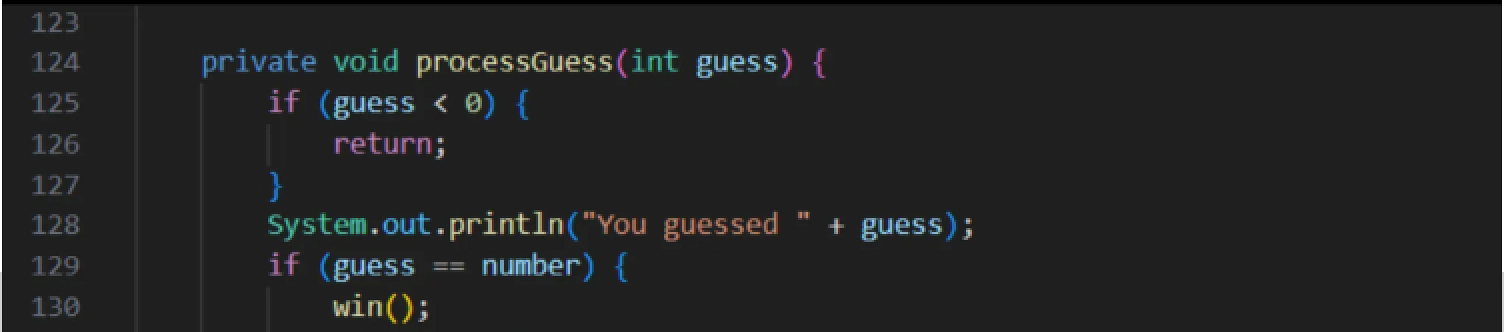
This screenshot shows the entire processGuess method.

Checklist Items (3)

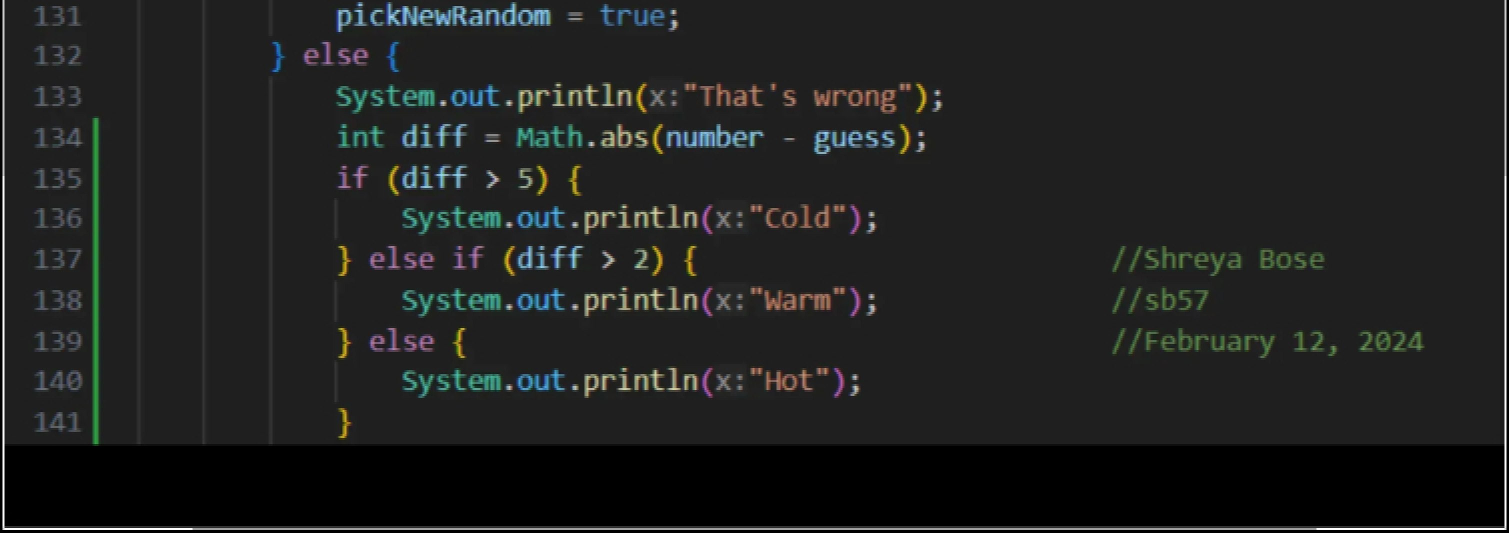
#2 Clearly caption the screenshot of what you're showing

#3 The code screenshot(s) clearly show the code specific to the feature

#4 A comment with the UCID/date is visible near the code change(s)



```
    private void processGuess(int guess) {
        if (guess < 0) {
            return;
        }
        System.out.println("You guessed " + guess);
        if (guess == number) {
            win();
```



```
                pickNewRandom = true;
            } else {
                System.out.println(x:"That's wrong");
                int diff = Math.abs(number - guess);
                if (diff > 5) {
                    System.out.println(x:"Cold");
                } else if (diff > 2) {                          //Shreya Bose
                    System.out.println(x:"Warm");               //sb57
                } else {                                        //February 12, 2024
                    System.out.println(x:"Hot");
                }
```

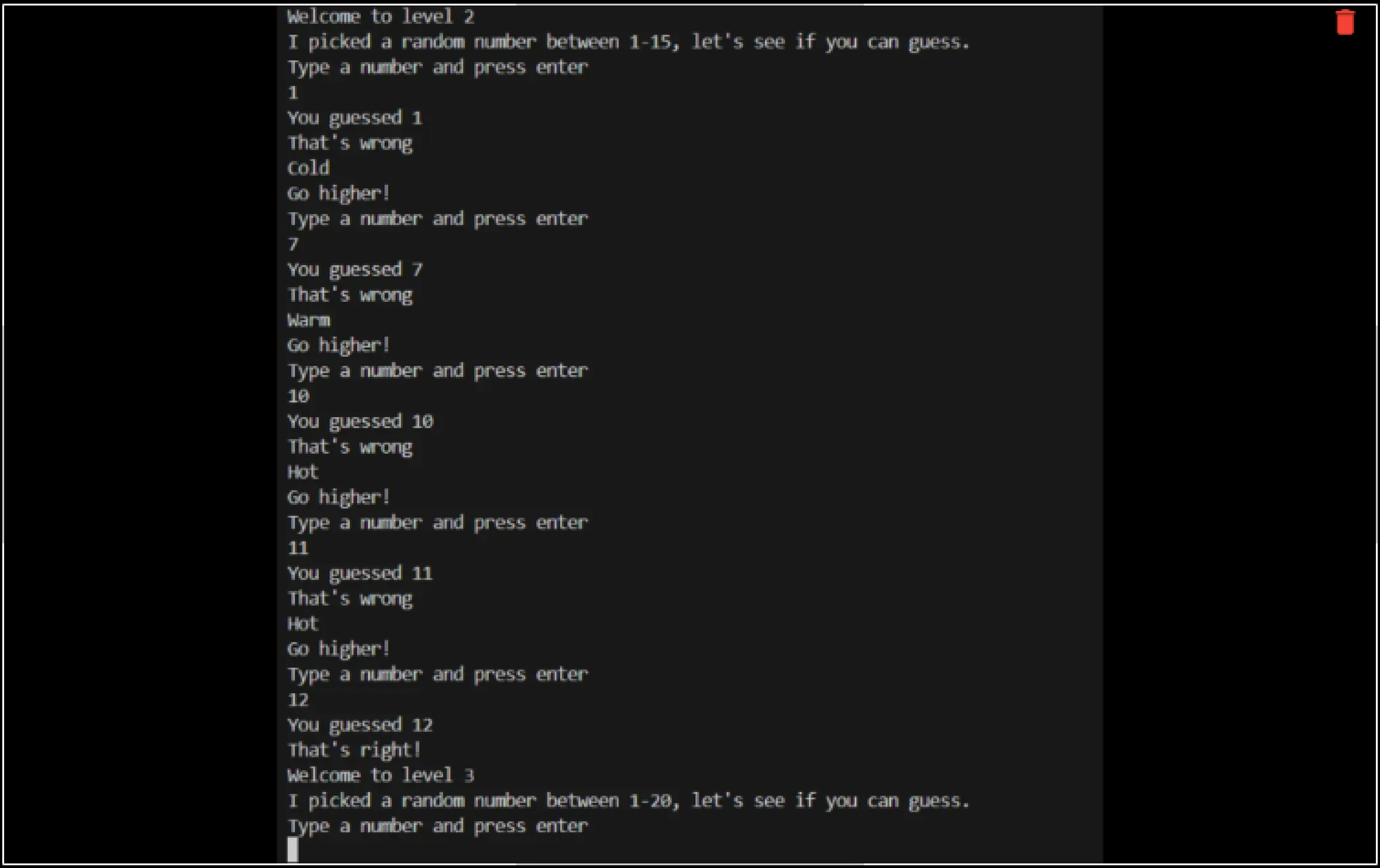
This screenshot specifically shows the snippet of the processGuess method which I edited in order to implement a hint on how cold, warm, or hot the incorrect guess is through an if/else if/else statement.

Checklist Items (3)

#2 Clearly caption the screenshot of what you're showing

#3 The code screenshot(s) clearly show the code specific to the feature

#4 A comment with the UCID/date is visible near the code change(s)



```
Welcome to level 2
I picked a random number between 1-15, let's see if you can guess.
Type a number and press enter
1
You guessed 1
That's wrong
Cold
Go higher!
Type a number and press enter
7
You guessed 7
That's wrong
Warm
Go higher!
Type a number and press enter
10
You guessed 10
That's wrong
Hot
Go higher!
Type a number and press enter
11
You guessed 11
That's wrong
Hot
Go higher!
Type a number and press enter
12
You guessed 12
That's right!
Welcome to level 3
I picked a random number between 1-20, let's see if you can guess.
Type a number and press enter
```

In this screenshot, I play the game in the terminal after the implementation. When my guess was too far away from the number, it told me I was cold, and the output changed as my guesses got closer to the number.

Checklist Items (2)

#1 Show implementation working by running the program

#2 Clearly caption the screenshot of what you're showing

● ^ COLLAPSE ^

Misc (2 pts.)

● ^ COLLAPSE ^

Task #1 - Points: 1

Text: Reflection

Checklist *The checkboxes are for your own tracking

#	Points	Details
□ #1	1	Example prompts: Learn anything new? Face any challenges? How did you overcome and issues?
□ #2	1	At least a few logical sentences related to the assignment.

Response:

I found the other prompts a bit challenging to implement. I tried but could not tell if I was working in the right direction. I think I was able to understand these implementations pretty well though.

● ^ COLLAPSE ^

Task #2 - Points: 1

Text: Pull Request URL

Details:
URL should end with /pull/# where the # is the actual pull request number.

URL #1
https://github.com/sb57-shreya/sb57-IT114-006/pull/5

● ^ COLLAPSE ^

Task #3 - Points: 1

Text: Waka Time (or related) Screenshot

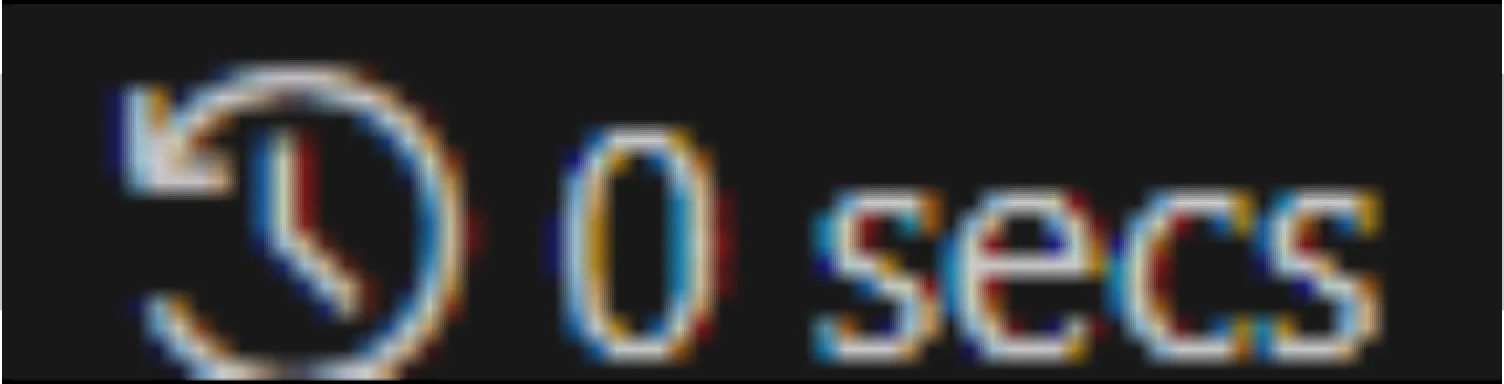
Checklist *The checkboxes are for your own tracking

#	Points	Details
□ #1	1	Screenshot clearly shows what files/project were being worked on (the duration of time doesn't correlated with the grade for this item)

Task Screenshots:

Gallery Style: Large View

Small Medium Large



I keep getting a "WakaTime working offline" message saying it will update when it is back online.

Checklist Items (1)

#1 Screenshot clearly shows what files/project were being worked on (the duration of time doesn't correlated with the grade for this item)

End of Assignment