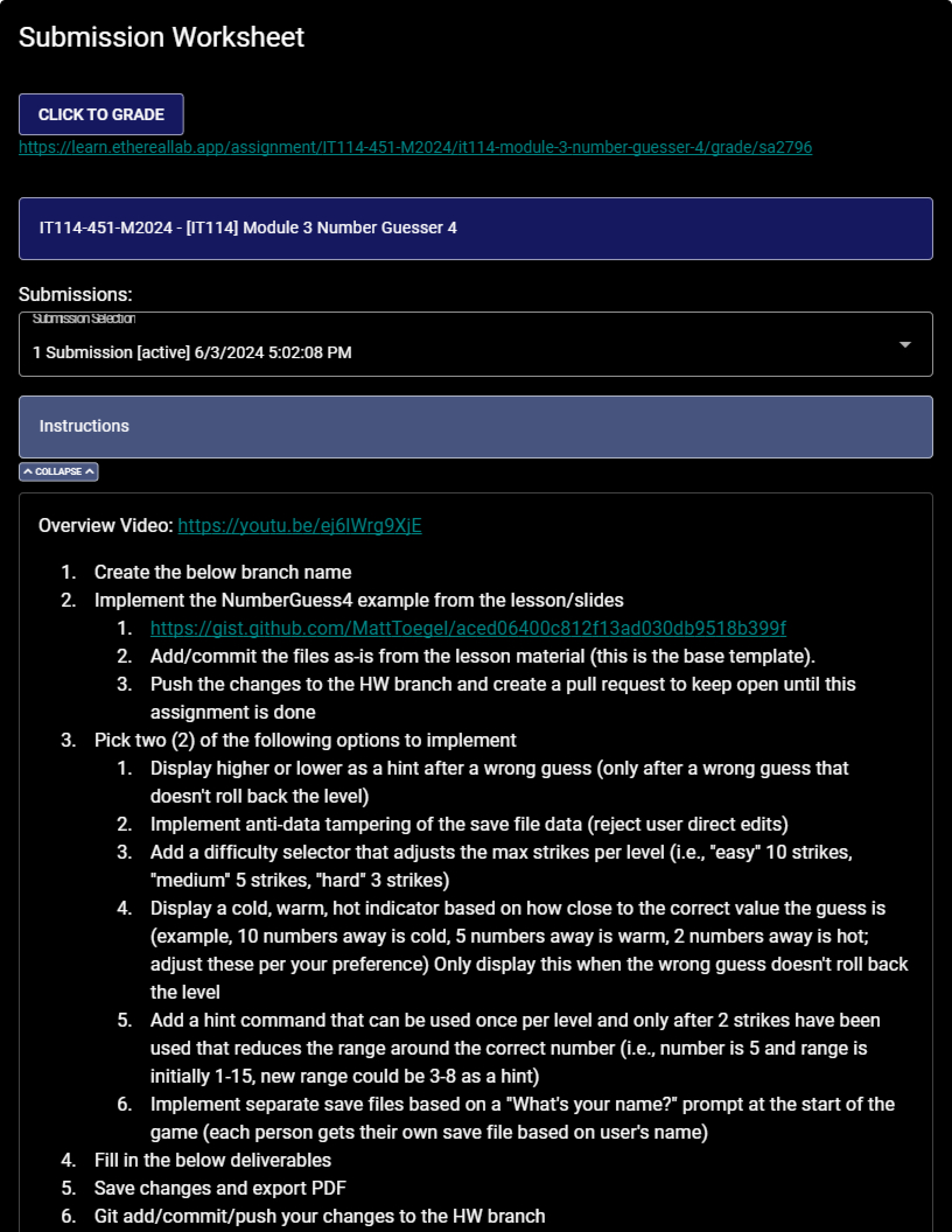


Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-451-M2024/it114-module-3-number-guesser-4/grade/sa2796

IT114-451-M2024 - [IT114] Module 3 Number Guesser 4

Submissions:

Submission Selection

1 Submission [active] 6/3/2024 5:02:08 PM

Instructions

COLLAPSE

Overview Video: https://youtu.be/ej6lWrg9XjE

1. Create the below branch name
2. Implement the NumberGuess4 example from the lesson/slides
    1. https://gist.github.com/MattToegel/aced06400c812f13ad030db9518b399f
    2. Add/commit the files as-is from the lesson material (this is the base template).
    3. Push the changes to the HW branch and create a pull request to keep open until this assignment is done
3. Pick two (2) of the following options to implement
    1. Display higher or lower as a hint after a wrong guess (only after a wrong guess that doesn't roll back the level)
    2. Implement anti-data tampering of the save file data (reject user direct edits)
    3. Add a difficulty selector that adjusts the max strikes per level (i.e., "easy" 10 strikes, "medium" 5 strikes, "hard" 3 strikes)
    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
    5. 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)
    6. 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)
4. Fill in the below deliverables
5. Save changes and export PDF
6. Git add/commit/push your changes to the HW branch

7. Create a pull request to main (if not done so before)
8. Complete the pull request (don't forget to locally checkout main and pull changes to prep for future work)
9. Upload the same PDF to Canvas

Branch name: M3-NumberGuesser-4

Tasks: 5 Points: 10.00

● Implementation 1 (4 pts.)

^COLLAPSE ^

● ^COLLAPSE ^

Task #1 - Points: 1

Text: Implementation Evidence

ⓘ Details:

Code screenshots must have ucid/date shown as a comment in the code.

Explanations must be your own words describing the logic and how the solution code solves the problem.

#1) Mention which option you picked and how you solved it

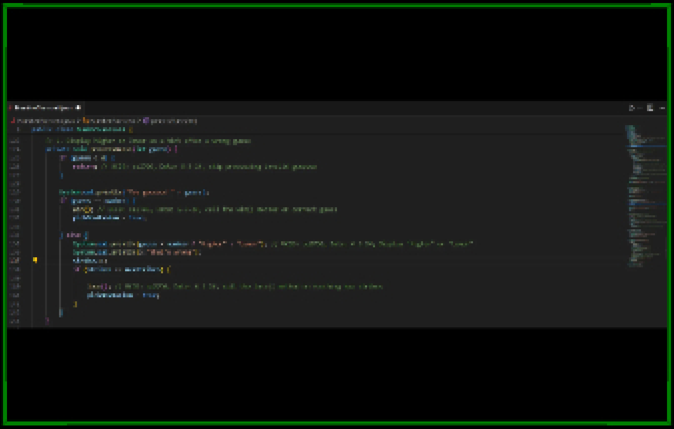
Explanation (required) ✓

Explain the logic of how you solved/implemented the chosen option (concrete details). Explain how the code works, don't just paste code snippets

PREVIEW RESPONSE

Item 1: Display higher or lower as a hint after a wrong guess (only after a wrong guess that doesn't roll back the level). This implementation shows if the player's guess is higher or lower than the correct number. Let's say that the guess is wrong, the codw will determine if the guess is higher or lower than the correct number (guess<number). Essentially if the guess is higher than the correct number it will output "higher" and if it is lower than it will output "lower".

#2) Add screenshots of the coded solution (ucid/date must be visible)

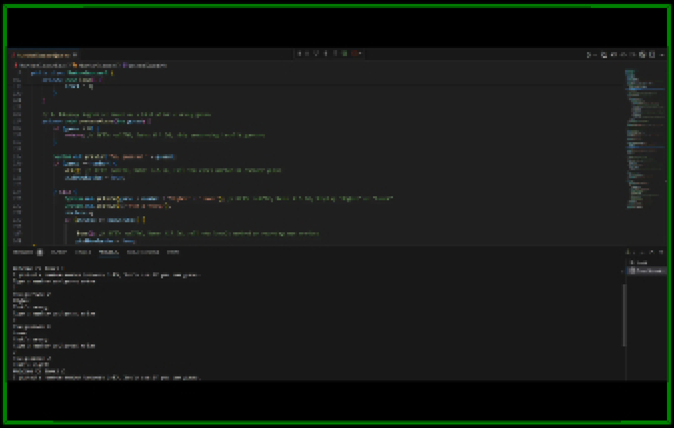


Caption (required) ✓

Describe/highlight what's being shown

Displaying the code for implementation 1

#3) Show implementation working by running the program



Caption (required) ✓

Describe/highlight what's being shown

Showing output of the code in terminal of Implementation 1.

● Implementation 2 (4 pts.)

^COLLAPSE ^

●

^COLLAPSE ^

Task #1 - Points: 1

Text: Implementation Evidence

ⓘ Details:

Code screenshots must have ucid/date shown as a comment in the code.

Explanations must be your own words describing the logic and how the solution code solves the problem.

#1) Mention which option you picked and how you solved it

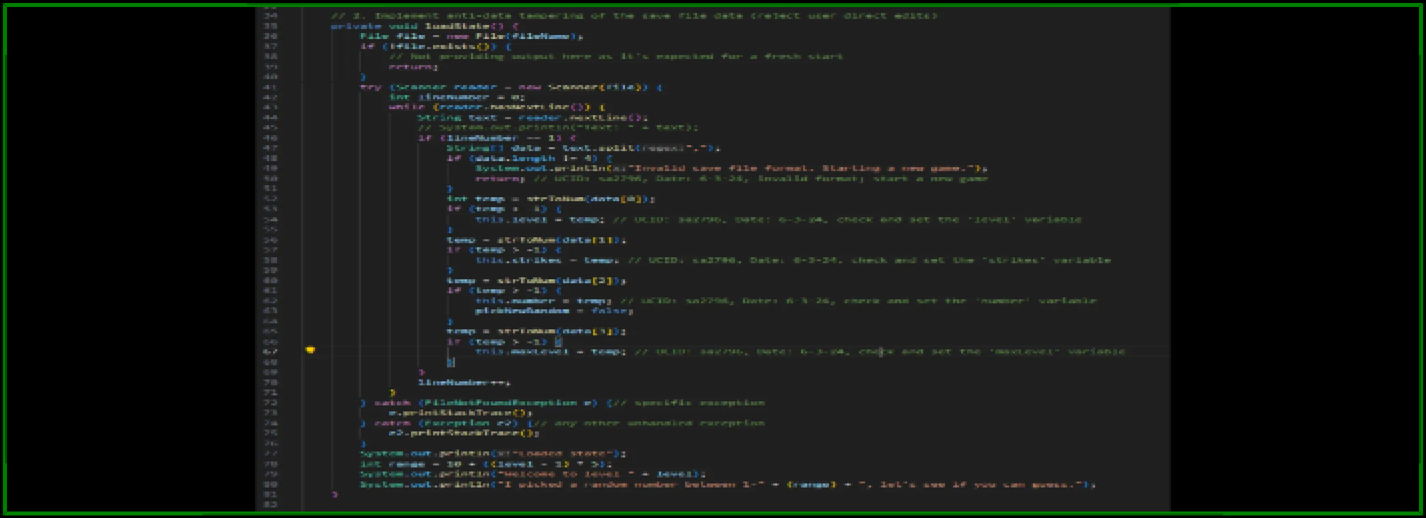
Explanation (required) ✓

Explain the logic of how you solved/implemented the chosen option (concrete details). Explain how the code works, don't just paste code snippets

PREVIEW RESPONSE

Item 2: Implement anti-data tampering of the save file data (reject user direct edits). This implementation checks if the format of save file aligns with the said structure. It uses if(data.length!=4). Let's say the format is wrong, tje code will take it as tampered data and starts a new game which makes it so users are not able to make unauthorized changes to save file.

#2) Add screenshots of the coded solution (ucid/date must be visible)



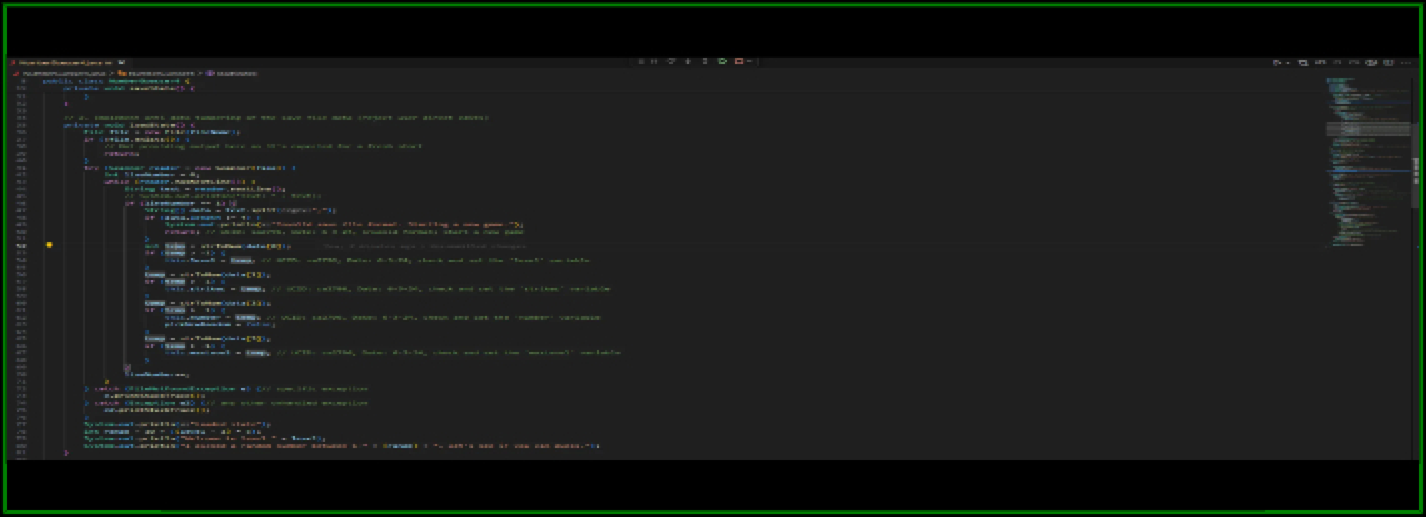
```
    // 2. Implement anti-data tampering of the save file data (reject user direct edits)
    private void loadState() {
        File file = new File(fileName);
        if (!file.exists()) {
            // Not providing output here as it's expected for a fresh start
            return;
        }
        try (Scanner reader = new Scanner(file)) {
            int lineNumber = 0;
            while (reader.hasNextLine()) {
                String text = reader.nextLine();
                // System.out.println("Text: " + text);
                if (lineNumber == 1) {
                    String[] data = text.split(regex:",");
                    if (data.length != 4) {
                        System.out.println(x:"Invalid save file format. Starting a new game.");
                        return; // UCID: sa2796, Date: 6-3-24, Invalid format; start a new game
                    }
                    int temp = strToNum(data[0]);
                    if (temp > -1) {
                        this.level = temp; // UCID: sa2796, Date: 6-3-24, check and set the 'level' variable
                    }
                    temp = strToNum(data[1]);
                    if (temp > -1) {
                        this.strikes = temp; // UCID: sa2796, Date: 6-3-24, check and set the 'strikes' variable
                    }
                    temp = strToNum(data[2]);
                    if (temp > -1) {
                        this.number = temp; // UCID: sa2796, Date: 6-3-24, check and set the 'number' variable
                        pickNewRandom = false;
                    }
                    temp = strToNum(data[3]);
                    if (temp > -1) {
                        this.maxLevel = temp; // UCID: sa2796, Date: 6-3-24, check and set the 'maxLevel' variable
                    }
                }
                lineNumber++;
            }
        } catch (FileNotFoundException e) {// specific exception
            e.printStackTrace();
        } catch (Exception e2) {// any other unhandled exception
            e2.printStackTrace();
        }
        System.out.println(x:"Loaded state");
        int range = 10 + ((level - 1) * 5);
        System.out.println("Welcome to level " + level);
        System.out.println("I picked a random number between 1-" + (range) + ", let's see if you can guess.");
    }
```

Caption (required) ✓

Describe/highlight what's being shown

Displaying the code for implementation 2.

#3) Show implementation working by running the program



Caption (required) ✓

Describe/highlight what's being shown

Showing output of the code in terminal of Implementation 2.

● Misc (2 pts.)

^COLLAPSE^

●

^COLLAPSE^

Task #1 - Points: 1

Text: Reflection

#1) Learn anything new? Face any challenges? How did you overcome any issues?

Explanation (required) ✓

Provide at least a few logical sentences

PREVIEW RESPONSE

This homework made me learn some concepts in java which were giving hints after wrong guesses and incorporating data checks for save files. These topics have amplified my knowledge on how to help a person's experience and protect data in certain applications.

● ^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/SHUAIB2796/sa2796-IT114-451/pull/6

● ^COLLAPSE ^

Task #3 - Points: 1

Text: Waka Time (or related) Screenshot

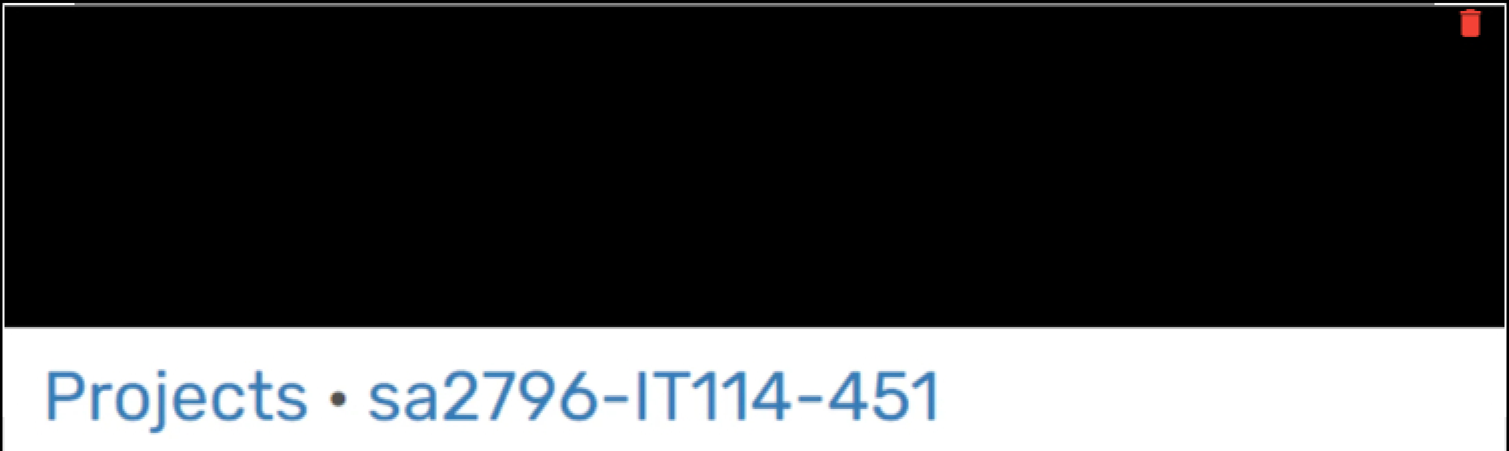
Checklist *The checkboxes are for your own tracking

#	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)

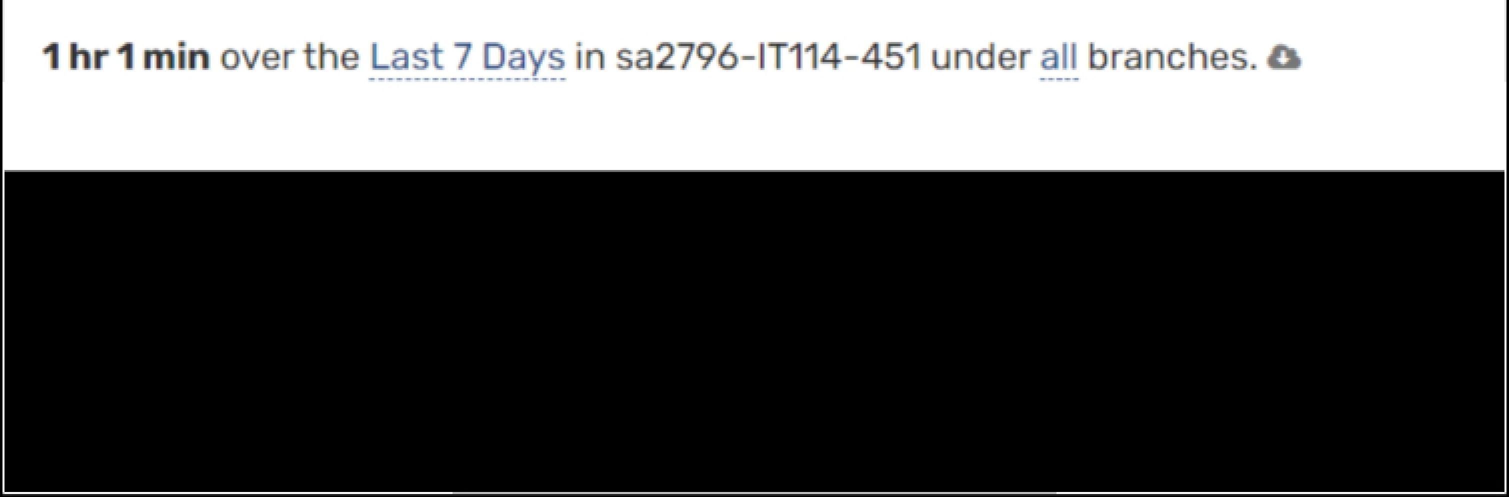
Task Screenshots:

Gallery Style: Large View

Small Medium Large



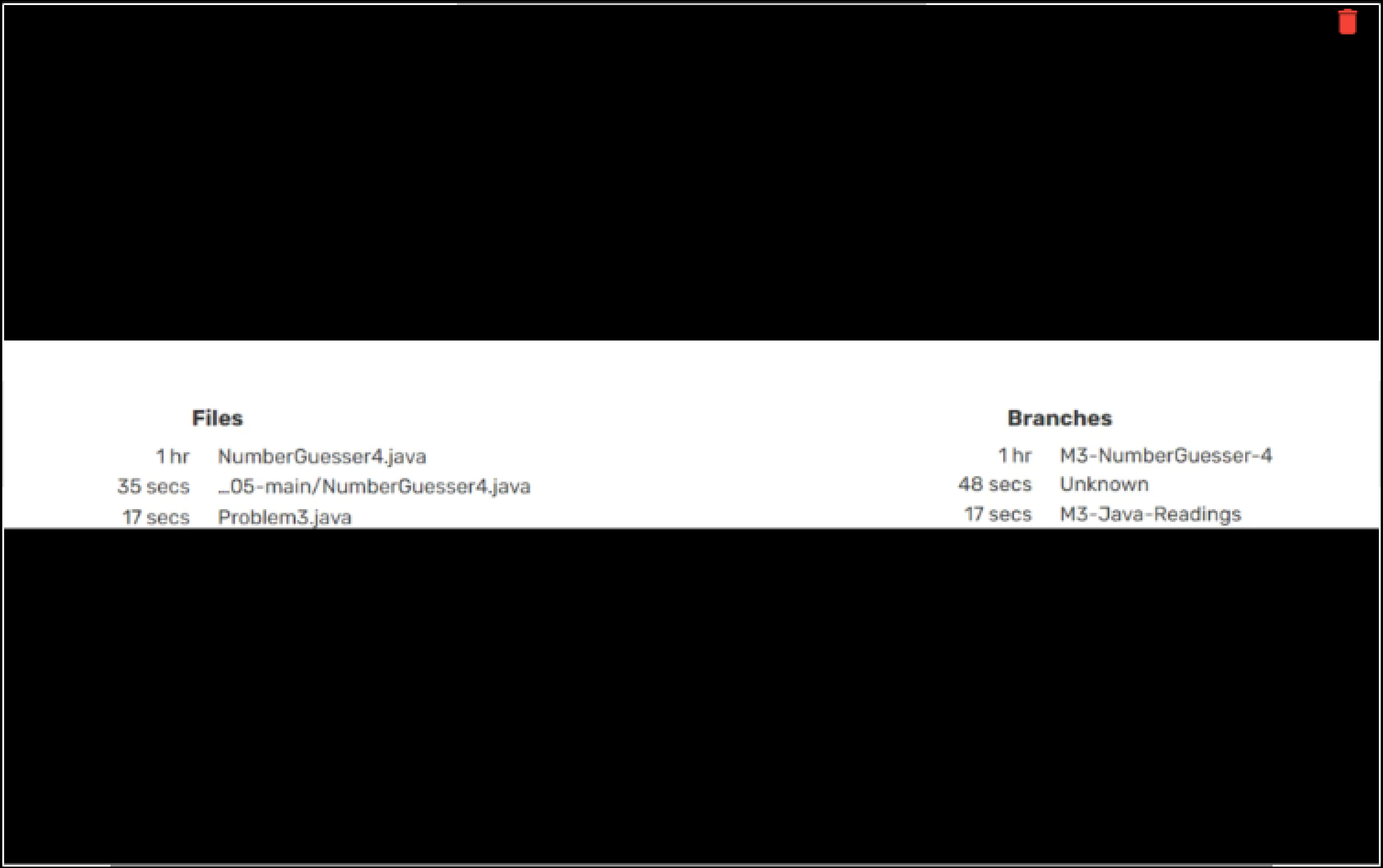
Projects • sa2796-IT114-451



1 hr 1 min over the Last 7 Days in sa2796-IT114-451 under all branches.

Overview

Checklist Items (0)



Files

1 hr	NumberGuesser4.java
35 secs	...05-main/NumberGuesser4.java
17 secs	Problem3.java

Branches

1 hr	M3-NumberGuesser-4
48 secs	Unknown
17 secs	M3-Java-Readings

Showing individual file times

Checklist Items (0)

End of Assignment