WEEK *8 REFLECTION

Week 8 Lab Report

Problem 4.1: Handling Basic Exceptions

- Predict

What do you think will happen if a user enters a string instead of an integer?

I predict that it will return “`That's not an integer. Try again.`”

RUN

Execute the program and observe the output. Did it match your expectations?

Yes. Inputting a string returned “`That's not an integer. Try again.″`

- Investigate

Run the program. What happens when you enter a string? What happens when you enter an integer?

When a string is entered, it returns the premade error message. When an integer is entered, it returns You entered: {{integer}}

Problem 4.2: Reading From a File

- Predict

What do you think will happen if the file “numbers.txt” does not exist?

- Investigate

Did it match your expectations?

Run the program. What happens if the file “numbers.txt” does not exist? What happens if the file exists and contains numbers?

Problem 4.3: Handling Multiple Exceptions

- Predict

What will the code snippet do?
It will print a sum of zero as nothing is in the file.

- Investigate

Did it match your expectations?
Yes, the code printed “sum of numbers: 0” when the code was run.

Discuss the different exceptions being caught in the try-catch blocks. Why might catching multiple exceptions be useful?
Using letters, symbols(: , #), assignment operators(+ = *), decimals, and percentage to test “File contains non-interger values”. To see the limits of the code and no unnecessary data slips though the try-catch block.

Problem 4.4: Copying a file

- Predict

Identify the errors in the code.
1.There are currently no txt files that can be referenced at the moment so it’ll hit a file not found exception.

2. scanner would only have one line so has next line shouldn't work

- Investigate

Identify any errors or exceptions that arise when you run the code.
Explain why they Occurred.
It is having a problem with the new word when creating a new printer and scanner

Group Reflection

- Facilitator

Engage all members of the group in a discussion of the questions below.
1. What are the primary differences between Scanner and PrintWriter in the context of file operations?
Scanner is SCANNING the incoming data into what its needed for
PrintWriter allows files to be written to from a program.

2. Why is it important to handle exceptions in file I/O operations?
Because if you don't there will issues with not being able to find out what the exact issue is

3. What is the significance of using a ‘try-catch‘ block when working with File I/O in Java?
It allows the errors to be read more easily and gives details of what is needed and or missing.

4. How does Java allow you to handle multiple exceptions? Why is this feature Beneficial?
Using Try-Catch allows you to check multiple exceptions at once making it easier to find errors

5. How can you handle multiple exceptions when reading from or writing to a file?
Using a try catch fo find if your missing a crucial piece

6. Why is it important to close resources like Scanner and PrintWriter after use?
So that it actually finishes scanning and writing if it isn’t closed it doesn’t work.

7. How can you ensure that resources like files are properly closed after operations, even in the face of exceptions?
Making sure to first close the file through the scanner /Printwriter.

8. In what scenarios, other than File I/O, do you think exception handling can be useful?
Using multiple loops to do something to make it easier to deal with an issue.

9. Reflect on a situation where not handling exceptions could lead to serious program

errors or data loss.
If we didn't find and eradicate the bugs the code would not work or could case a crash

- Spokesperson

The spokesperson should answer the following questions, then seek feedback from the rest of the group.

Did the group seek help when stuck on a problem? If so, about what and how was the problem resolved? If not, how was communication among group members, what was good, and what can be improved?
Yes, one of the group members didn't know how to make the .txt file so the group provided multiple ways to make a .txt file in the terminal. "vim numbers.txt" and "touch numbers.txt".

Be ready to discuss these questions with the TA.

- Quality Control

Quality Control should answer the following question, then seek feedback from the rest of the group.

How can the solution(s) we created today be improved?

These solutions can be improved by separating the functions into individual methods. A lot of the functions we wrote are not well organized into methods, so by doing so, we can significantly improve maintainability and scalability. Additionally, we can implement test cases to better diagnose what features our code may be missing.

Note: Improvements do not need to add functionality or features, think about maintainability, understandably, and testing as alternate avenues towards improved code.

- Process Analyst

Process Analysts should answer the following questions, then seek feedback from the rest of the group.

What challenges happened during the lab that halted progress? How can we address these challenges in the future?

Important! All students should be collecting pieces of evidence for their end of the

semester reflection. Be ready to discuss this questions with the T