What did you learn?

I did not learn anything new from this lab. However, it has generated questions that could lead to new learning opportunities. For the most part, this lab reviews information learned in CMSC203 thus most of it was largely information recall.

What issues did you encounter, if any?

I did not encounter any significant issues. Although, whilst completing this assignment It did generate a couple of questions. I have included a questions section with the questions that came up.

What would you have done differently?

this lab was fairly small and very straight foward. Hence, I do not think I would change the way I implemented the specifications.

How can you apply this concept in the future?

Both of the concepts discussed/implemented in this lab are extremely useful across a wide range of Java applications, and computer science in general for that matter. Prior to utilizing custom exceptions, we had very limited options in our “tool belt” per se to deal with unusual and unexpected problems. Custom exceptions open a very wide range of solutions for unexpected circumstances.

JUnit provides a good avenue for testing which we had implemented extensively in CMSC203. It is very useful for testing any class or program. However, sometimes it provides limited information when something fails. The assertTrue method only notifies the programmer that “true” was expected but “false” was obtained. When testing a string this provides no info on why it failed. Nonetheless, JUnit is more efficient than performing test cases by hand, especially for larger projects.

Anything else you would like to share?

I cannot think of anything I would like to share at this moment. If I think of something I will be sure to add those ideas to the discussion forum.

Questions:

- How do we determine whether our custom exception should be checked or unchecked?
    - if we decide it should be unchecked do we extend RunTimeException, and extend Exception otherwise?
- if an unchecked exception is thrown will the program always terminate or is it conditional upon what exception is thrown?
    - I likely knew the answer to this question when I was taking CMSC203 but it has been a while.
- if the isValid method throws an exception what is the point in returning false? Throwing an exception means that the SSN is invalid and once it is thrown the “return false;” statement will not execute. This seems to be redundant. Both things provide the same

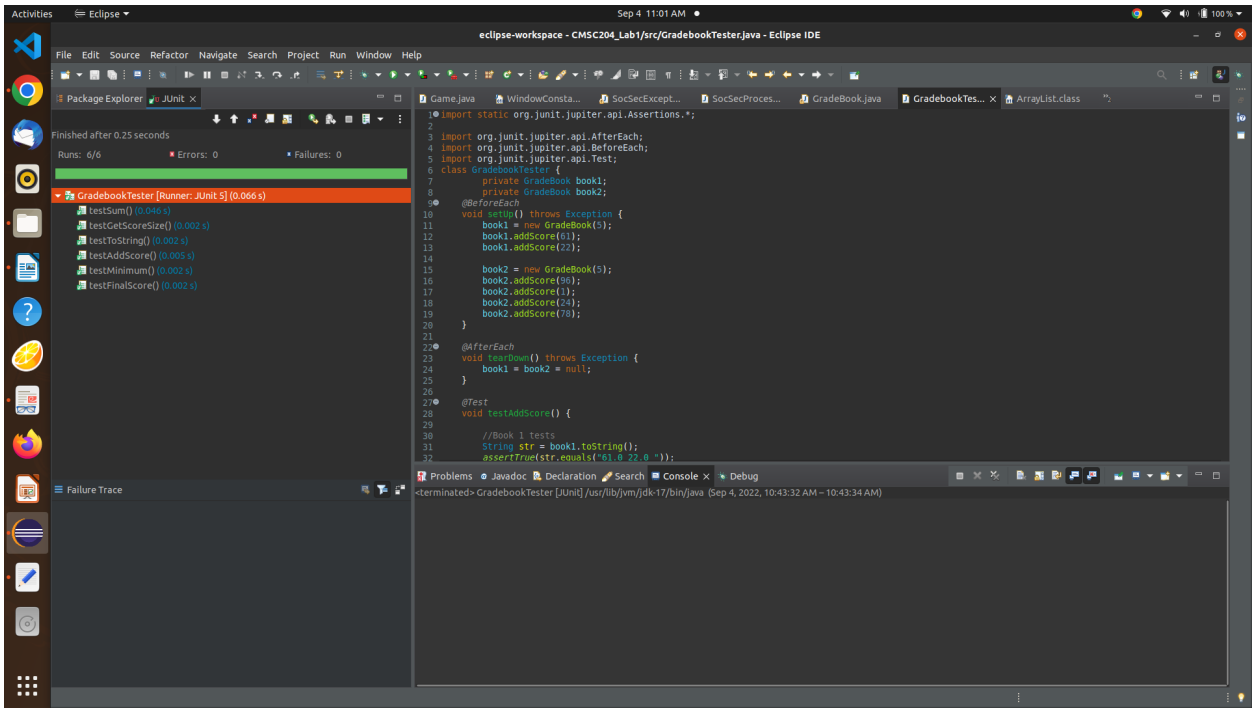
information. Of course, the exception, in the way we are using it, returns false with additional features/information.
    - is my analysis wrong? is there a reason to return true AND throw an exception for an invalid case?

Test Cases:

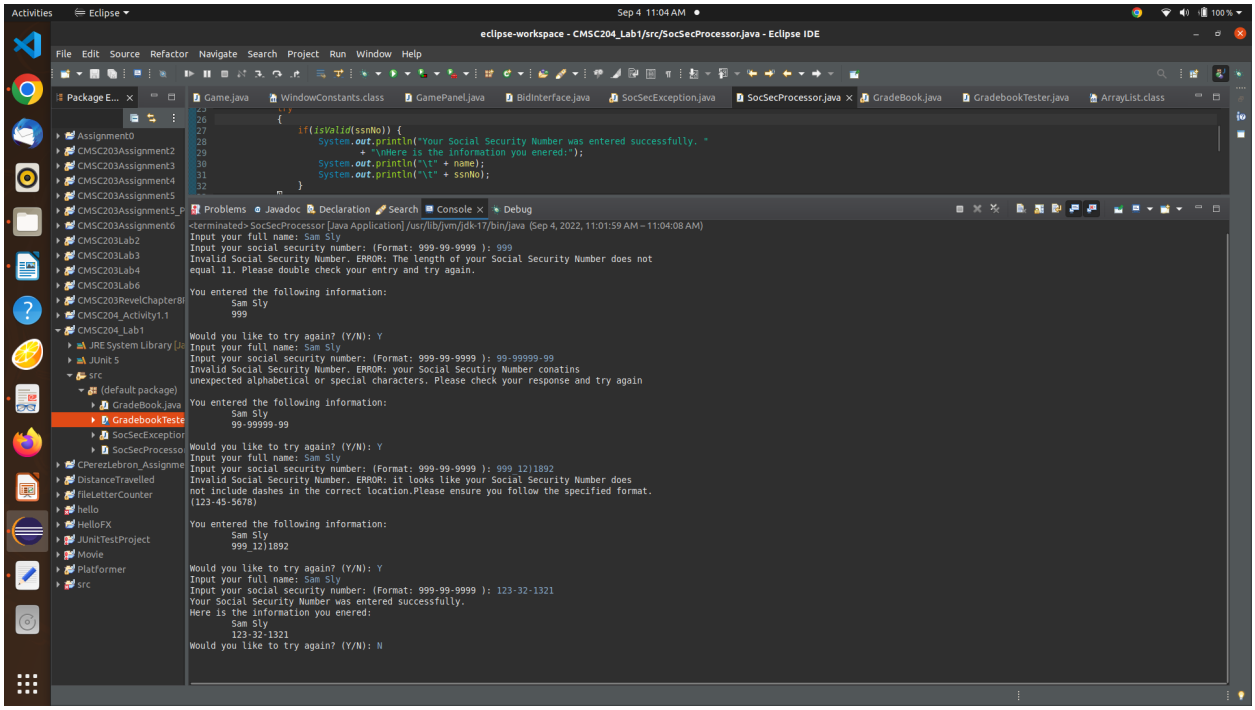
- to check length constraint:
    - 9
        - should fail far too short
- to check that digits are where they should be and not any other character
    - 99-99999-99
        - should fail position with index 2 is not a digit (it is a dash)
- to check that a hyphen is in the current position and nothing else
    - 999_12)1892
        - should fail because instead of dashes other characters are used

Screenshots of my GitHub repo

Screenshots of JUnit test runs



Screenshots of Exception test runs


Activities
Eclipse
Sep 4 11:04 AM
eclipse-workspace - CMSC204_Lab1/src/SocSecProcessor.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package E...
Game.java
WindowConstants.class
GamePanel.java
BidInterface.java
SocSecException.java
SocSecProcessor.java
GradeBook.java
GradebookTester.java
ArrayList.class
Assignment0
CMSC203Assignment2
CMSC203Assignment3
CMSC203Assignment4
CMSC203Assignment5
CMSC203Assignment5_P
CMSC203Assignment6
CMSC203Lab2
CMSC203Lab3
CMSC203Lab4
CMSC203Lab6
CMSC203RevelChapter8
CMSC204_Activity1.1
CMSC204_Lab1
JRE System Library
JUnit 5
src
(default package)
GradeBook.java
GradebookTeste
SocSecExceptior
SocSecProcesso
CPerezLebron_Assignme
DistanceTravelled
FileLetterCounter
hello
HelloFX
JUnitTestProject
Movie
Platformer
src
26 {
27 if(isValid(ssnNo)) {
28 System.out.println("Your Social Security Number was entered successfully. "
29 + "\nHere is the information you enered:");
30 System.out.println("\t" + name);
31 System.out.println("\t" + ssnNo);
32 }
Problems Javadoc Declaration Search Console Debug
<terminated> SocSecProcessor [Java Application] /usr/lib/jvm/jdk-17/bin/java (Sep 4, 2022, 11:01:59 AM – 11:04:08 AM)
Input your full name: Sam Sly
Input your social security number: (Format: 999-99-9999): 999
Invalid Social Security Number. ERROR: The length of your Social Security Number does not
equal 11. Please double check your entry and try again.
You entered the following information:
Sam Sly
999
Would you like to try again? (Y/N): Y
Input your full name: Sam Sly
Input your social security number: (Format: 999-99-9999): 99-99999-99
Invalid Social Security Number. ERROR: your Social Secutiry Number conatins
unexpected alphabetical or special characters. Please check your response and try again
You entered the following information:
Sam Sly
99-99999-99
Would you like to try again? (Y/N): Y
Input your full name: Sam Sly
Input your social security number: (Format: 999-99-9999): 999_12)1892
Invalid Social Security Number. ERROR: it looks like your Social Security Number does
not include dashes in the correct location.Please ensure you follow the specified format.
(123-45-5678)
You entered the following information:
Sam Sly
999_12)1892
Would you like to try again? (Y/N): Y
Input your full name: Sam Sly
Input your social security number: (Format: 999-99-9999): 123-32-1321
Your Social Security Number was entered successfully.
Here is the information you enered:
Sam Sly
123-32-1321
Would you like to try again? (Y/N): N