CSC171 — Homework 15

Exceptions

The goal of this assignment is to give you experience with Java Exceptions. Be sure to include a descriptive comment about the program at the start of the main class, and descriptive comments for any methods you define.

Questions

1. Write a program that reads a string from the user, converts that string to an integer using `Integer.parseInt`, increments that value and prints the result. Handle the `NumberFormatException` to print an appropriate message if the user's input is not the decimal representation of a number.

2. Write the beginnings of a simple calculator program. Your program should read a number, then a string, then another number from the user (all on one line). If the string is "+" you should add the two numbers and print the result. If it's "-" you should subtract them, and so on for "*", "/", and "%". Catch both `ArithmeticException` and `InputMismatchException` to prevent crashes and print suitable (different) messages on bad input.

3. Write a program with a static method that takes a `Scanner` as argument and returns the next integer read from the `Scanner`. If the user doesn't enter a number, you should re-prompt and try again. To do this, catch any `InputMismatchException` and use that in combination with a loop to re-prompt and re-read. You will need to use `Scanner.next` or `Scanner.nextLine` to skip the non-numeric input. (Note that you could also use `Scanner.hasNextInt` for this, but that's not the point of this exercise.) Include code to demonstrate the use of your method.

4. Write a program with a static method that takes a `List` of `Object`s and an integer index as parameters and returns the element in the `List` at the given index or `null` if there is no such element. Do this by catching the `IndexOutOfBoundsException`. You could also do this with a conditional, right? Think about which is better, when, and why. Include code to demonstrate the use of your method (I recommend using a `List` of `Integer`s for the test case, but do whatever you want).

5. Write a program that prompts the user for a filename and prints the contents of the file to the console. If the file does not exist, print an informative message.

Grading Scheme

Equal weight for each part.

Doesn’t compile or is trivial	$<$ 50%
Compiles and is non-trivial	$\geq$ 50%
Complete and correct with good style and comments	100%
Incomplete, incorrect, bad style, no comments	$<$ 100%

Submission Requirements

Your submission **MUST** include a file named “`README.txt`” with your name, your NetID, the assignment number, and your lab section. This file should explain anything we need to know about how to build and run your project. In particular, be sure to explain how to run what parts of your submission for each question in the assignment.

Submit your solution as a single ZIP archive to BlackBoard before the deadline.

Late homeworks will not be graded and will receive a grade of 0.

All assignments and activities associated with this course must be performed in accordance with the University of Rochester’s Academic Honesty Policy.