CSC 360 Assignment #1

Total Points: 2 x 20 = 40 points

Submission

- Submit only your *.java* source files on Canvas.
- Name the source files in the format *LastnameFirstnameA1P1.java* and *LastnameFirstnameA1P2.java*,
- You will replace *LastnameFirstname* with your own last and first name.

Problem 1 (*InputMismatchException*)

Review the *InputMismatchExceptionDemo.java* source code from the lecture. Write a program that prompts the user to read in two integers and then displays their sum.

The behavior of the program should be as follows:

- Give a separate prompt for each integer.
- Prompt the user to enter a number until the number's format is correct.
- If the user enters the first integer correctly, and then enters the second integer incorrectly, the program should prompt the user to enter only the second integer again, and not start from the first integer again.
- Follow the sample output to include the necessary texts for the user prompts and other outputs.

The following is a sample run indicating how your program should behave.

```
Enter first integer: hi
Try again. (Incorrect input: an integer is required)
Enter first integer: six
Try again. (Incorrect input: an integer is required)
Enter first integer: 6
Enter second integer: five
Try again. (Incorrect input: an integer is required)
Enter second integer: 5
The sum is 11
```

Problem 2 (*Process values in a text file*)

Review the *ReadData.java* source code from the lecture. Write a program that prompts the user for the name of a text file. The program will then read the contents of the file and report accordingly.

The behavior of the program should be as follows:

- If the file cannot be found, then the program will print a message to that effect and terminate by executing System.exit(1).
- The file should consist of a sequence of integers, one integer per line.
- The program will read in each line (using *nextLine()*), parse it as an int (using *Integer.parseInt()*).
- The program will calculate the average of the integers read from the file. The average should be computed as a double.
- The program should handle *NumberFormatException* for bad input lines.
- The program will report the inputs being read from the file which are not integers. Every time the program reads in a line that cannot be parsed (i.e. *Integer.parseInt()* throws a *NumberFormatException*) the program should print an error message with the bad line, and then continue reading the rest of the lines.
- The final output should display the number of parsable (valid input) lines, the average of the parsable values as a double, and the number of unparsable (invalid input) lines.
- Follow the sample output to include the necessary texts for the user prompts and other outputs.
- For your convenience, create a text file with sample input lines, and place it in the project directory.

Let us assume you have a file *numbers.txt*, with the following contents:

```
5
15
312
16
eight
seven
44
eighty-five thousand and sixty-two
13 98
93
```

The following are two sample runs indicating how your program should behave.

Run 1:

```
Enter name of input file: numbers.txt
Cannot parse eight as an integer.
Cannot parse seven as an integer.
Cannot parse eighty-five thousand and sixty-two as an integer.
Cannot parse 13 98 as an integer.
Number of parsable lines: 6
Average value: 80.83333333333333
Number of unparsable lines: 4
```

Run 2:

```
Enter name of input file: abcd.txt
Could not find file: abcd.txt
```