Lab 7: File I/O and Exceptions

UNK CSIT 150 Object Oriented Programming

Objectives

- Practice reading a text file
- Practice reading a binary file
- Practice parsing a string
- Practice dealing with exceptions

General Requirements

In this lab, you must write your code following the proper programming style. The bottom line is:

- Use *indentation* to show the logical structure of your code
- Use blank lines to separate code blocks and give each code block a comment
- Give documentation comment to the class and each method

Programming Practice

1. A binary file named *doubles.dat* is provided with this lab. This file has a list of numbers (doubles) such as 10.2, 20, 5.1, 8, 9.9. This is a binary file, so it is not readable. It does not contain the commas between the numbers. We do not know how many numbers are in the file.

   Read the numbers stored in the binary file into an array. Output the array of values and calculate the average of the numbers in the array.

2. **Bonus**. A text file named *doubles.csv* is provided with this lab. This file has a list of numbers (doubles) in a single line of data in a form of 10.2, 20, 5.1, 8, 9.9. All of the numbers are separated by commas (,) but we do not know how many numbers are in the file.

   Read the numbers stored in this comma-separated value (csv) text file into an array. Output the array of values and calculate the average of the numbers in the array.

Other requirements

- **Use exception handling appropriately.**
- Make your solution as modular as possible, and re-use methods appropriately.

Lab 6: Binary File/Exceptions

Name(s): ____

Evaluation

Requirement	**Possible Points**	**Points Received**	**Comments**
Proper use of Exceptions. The input file may not exist, may be empty, may have the wrong kind of data.	3		
Reading from an input file, and storing to a properly sized array. The file provided is just a sample file. We may use a file with more or less numbers when grading.	4		
Output data	1		
Calculate and output average	2		

Programming style	**Possible Points**	**Points deducted**	
Inconsistent indentation	**-1**		Indent at least 3 spaces inside each brace
Poor use of white space	**-1**		Too many or too few lines between statements
Heading documentation, includes programmer names, date, algorithm, basic purpose	**-2**		
All sources not cited (Remember to cite all code used, even class demo code.)	**-1**		
JavaDocs not used on each method	**-1**		
Poor variable names	**-1**		Should not start with uppercase. No single letter variables.
Poor structure/logic issues	**-2**		
Program specifications			
Bonus:			