Rochester Institute of Technology
National Technical Institute for the Deaf
Information and Computing Studies Department

Name: _____

NACA.161 Programming Fundamentals II
In-class Exercise #26 – File IO - Writing

Overview

This exercise is designed to let you read and write files using various IO classes that read and write text files.

Writing data to a file

1) Create a class called **ReadWrite** that contains a main method.

2) In the main method, create an object that will let you write to a file called **out.txt**

   What does this line look like?

   FW = new FileWriter("out.text");

   For now, only enter this line.

3) Compile the code.

   What error did you get?

   error: cannot find symbol

4) Correct the error by catching the exact exception type reported. Print the output of the **toString** method inside the catch block.

   What exception did you need to catch?

   Unreported exception IO Exception

5) Create a string with the words: "This is the first file I am writing"

6) Write this string to the file.

What method did you use?

fw.Write(s)

7) Create the another string with words "This is line number two"

8) Now write this string into the file after the first string you wrote. Compile and run.

What was the output?

ThisisfirstfileIamwritingThisisnumbertwo

Obviously these 2 strings did not print on different lines. How can you correct this problem?

By adding a new line char

What is the character to write a newline? \n

9) Add another write statement that will write a newline between the 2 existing write statements.

10) Close the file.

11) Compile and run. Correct your program until it writes two separate lines to the file. Remember since this is a text file, you can open it with textpad, notepad, or JGrasp.

Reading the file that you created

12) One way to see the data in the file is to read each character in the file and print it. Add code that opens the file you just created for reading. Create another try/catch statement to hole the code in this step and the following steps to read the file.

What class did you use?

ReadWrite

Compile the class to make sure your code still works.

13) Now add code that reads the entire file and prints the results to the screen. Close the file after reading all of the data. Compile and run the program. You should now see the contents of the file on the screen.

Adding more data into the file

14) Modify the portion of the program that writes the data to write 25 lines of the form:

```
This is line
1 This is
line 2
...
This is line 25
```

Compile and run the program until can write the 25 lines and then read and print them correctly.

Converting digits into a integer number

15) Remove your code to write the 25 lines and now write one line where the symbol "^" represents a space:
^^^1527^^
Compile and run the program. Correct any errors until the program works correctly.

16) Modify the loop to append each character to a string. Then print the string. Compile your program until it can correctly print the string.

17) We want to remove the leading and trailing spaces in the string. Find a method in the String class which will do this. Then print the new string. Write this new code after the try/catch statement to read the file. Compile your program until it can correctly print the string.

18) Convert the contents of the string into an integer, add 450 to it and print the sum. Hint: use the Integer class.

Signoff

When you complete all of the steps successfully and answer all of the questions, contact your instructor to check if your application(s) executes correctly and to review your code. We will initial the line below.

_____ Successful execution of code

If you do not finish the program during the class period, contact your instructor to check to review your code and initial below.

_____ Code not completed during lab time

You may then submit your work at the start of next class. You may not use the work period of the next class to complete this assignment. If you do not have a signature, then you cannot receive any points for this assignment.