ICS4U Module 5: Note ↓ Exercise 2c

The File Streams

- A *stream* processes characters
- The file stream keeps track of the *file position*
    - The point where reading or writing last occurred
- The file streams are used to perform *sequential file access*
    - All reading and writing is performed one character after another or one line after another

In a word processor, this file looks like:

Drew 84
Tia 92

Cr: carriage return character
Lf: line terminator
-1: end of file

Visualization of Stream of Data

D	r	e	w		8	4	Cr	Lf	T	i	a		9	2	Cr	Lf	-1

Class FileReader (java.io) FileReader creates an input file stream.

Method	Description
Constructor: `FileReader(File fileName)`	Creates an input file stream for the File object. This constructor throws a FileNotFoundException if the file does not exist
`close()`	Closes the input file stream. This method throws and IOException if the file cannot be closed

Class BufferedReader (java.io) BufferedReader reads text from the stream.

Method	Description
Constructor: `BufferedReader(Reader stream)`	Creates a buffered-input stream from stream. Reader is the FileReader superclsas
`read()`	Reads a single character from the input stream. This method throws and IOException if the stream cannot be read
`readLine()`	Reads a line of text from the input stream. This method throws an IOException if the stream cannot be read
`close()`	Closes the input file stream. This method throws an IOException if the stream cannot be closed

Sample Program: prompts the user for names and scores and then writes them to a new file.

```
/*
 * ReadFile.java from Module 5
 * A program that demonstrates reading from a file.
 */

 import java.util.Scanner;
 import java.io.*;

 /**
  * A program that displays the contents of a file.
  */
public class ReadFile {

      public static void main(String[] args) {
            File textFile = new File("operating_system.txt");
            FileReader in;
            BufferedReader readFile;
            String lineOfText;

            try {
                  in = new FileReader(textFile);
                  readFile = new BufferedReader(in);
            while ((lineOfText = readFile.readLine()) != null ) {
                  System.out.println(lineOfText);
            }
            readFile.close();
            in.close();
      } catch (FileNotFoundException e) {
                  System.out.println("File does not exist or could not be
found.");
                  System.err.println("FileNotFoundException: " +
e.getMessage());
            } catch (IOException e) {
                  System.out.println("Problem reading file.");
            System.err.println("IOException: " + e.getMessage());
      }
      }
 }
```

Note: a try may include multiple catch statements (both the FileReader and BufferedReader throw exceptions)

The close() methods must be closed in reverse order that they were opened

Programming Exercise:

Create an Assign application that reads and then displays the contents of a file containing instructions for this assignment. Use Notepad or some other word processor to create the file. Be sure that the file is saved as a Text file (TXT). The Assign application will need to include the correct path to the location of the file. If a path is not specified, the file must be placed in the same folder as the Assignment executable file.

Submit your source code to the Google Doc "ICS4U – Activity Submission Form"