CSCI 2302

Text I/O Lab

Contents

[Purpose 1](#_Toc93386524)

[Learning Goals 1](#_Toc93386525)

[Notes 1](#_Toc93386526)

[Task 4](#_Toc93386527)

[Sample Run 4](#_Toc93386528)

[Submit 5](#_Toc93386529)

Purpose: Data on non-volatile storage media is stored in named locations on the media called files. By reading and writing files, programs can save information between program runs.

Working with files is a lot like working with a notebook. To use a notebook, it has to be opened. When done, it has to be closed. While the notebook is open, it can either be read from or written to.

Learning Goals: To gain the knowledge of reading and writing to test files so that we can apply this knowledge to future programs.

Notes:

* Declaring and instantiating a File object creates a way we can handle a file. We can use that file to test if it exists, if we have permission to read from the File, and/or permission to write to the File.
* Once we are finished with the resource (the File object), we want to be able to close the resource.

/\* Java version 8

\* This program is an example and explanation for handling text IO (text input and text output)

\* You will only need to use the Scanner object and the PrintWriter object for handling text IO

\*/

import java.io.\*;

Make sure to add this to the methods that do IO

import java.util.\*;

public class FileIOExample {

public static void main(String[] args) throws IOException {

System.out.println("This program is an example of how to read from a file and how to write to a file");

System.out.println("To read from a file you need to declare that file as an object for the program to be used. ");

System.out.println("This can be accomplished in two ways.");

System.out.println("First way: declare the File object, the set the Scanner to that File object");

// creating a File object

File inFile = new File("textFile.txt");

System.out.println("If you choose this option, you WILL be able to do ALL of the methods that go with a File object; i.e. exists(), canRead(), and others.");

// setting the Scanner to read from the file object

Scanner input = new Scanner(inFile);

System.out.println("Second way: create the Scanner to read from a 'on-the-fly' created objected File object");

// creating a Scanner to read from a File object that was created on-the-fly

Scanner inputAnother = new Scanner(new File("textFile.txt"));

System.out.println("If you choose this option, you will NOT be able to do any of the methods that go with a File object;i.e. exists(), canRead(), and others.");

System.out.println("Once the Scanner has been set to read from the file, you use the Scanner as you would normally reading content from the keyboard.");

System.out.println("A loop to read all the contents of a file");

while(input.hasNext()){

// reading in a word and printing it right away

System.out.println(input.next());

}

System.out.println("Once the file has been read from, you close the resource up");

// close the Scanner that is reading from the file

input.close();

System.out.println("To write to a file:");

System.out.println("Writing to a file requires an object that can write - the PrintWriter");

System.out.println("When you create a PrintWriter, it needs to know the file that it is writing to");

System.out.println("To create a PrintWriter & the File object:");

System.out.println("Once again there are two ways to accomplish this.");

System.out.println("First way: declare the File object, the create the PrintWriter to that File object");

// Creating a File to write to

File outFile = new File("textWrittenToFile.txt");

// creating a PrintWriter object that will write to that file

PrintWriter out = new PrintWriter(outFile);

// FYI - I give the identifier out to my PrintWriter object as I am already in the habit of writting: out.println("writing words");

System.out.println("If you choose this option, you WILL be able to do ALL of the methods that go with a File object; i.e. exists(), canWrite(), and others listed.");

System.out.println("Another way - is to create the file on the fly ");

PrintWriter outAnother = new PrintWriter(new File("anotherTextWrittenToFile.txt"));

System.out.println("If you choose this option, you will NOT be able to do any of the methods that go with a File object i.e. exists(), canWrite(), and others.);

System.out.println("You use the PrintWriter just as you normally write to the screen – except you do NOT need to add System.out");

System.out.println("Another loop to read all the contents of a file");

while(inputAnother.hasNext()){

// reading in a word from a file and printing it right away to a file out.println(inputAnother.next());

}

System.out.println("Once the file has been read from, you close the resource up");

// close the Scanner that is reading from the file

inputAnother.close();

System.out.println("Once the file has been written to, you close the resource up");

// close the PrintWrite that is writing to the file

out.close();

System.out.println("Since our programs in this class are so small, you will probably not see anything written to file if you do not utilize the close() method.");

System.out.println("The operating system only does things when it needs to as it is so busy doing stuff and will not waste its resources when not necessary - hence the output just sits there in RAM, not written to the file.");

System.out.println("To combat this, you can use the method flush(). This method informs the os that it has to write the contents now.");

System.out.println("You must always close the PrintWriter though!");

}// end main

}// end class - FileIOExample

Task: Implement a program, MySFAUsername\_textIO\_Lab.java, that will:

* Declare and create a Scanner object to read from ai.txt
* Declare and create a PrintWriter object to write to out.txt
* Define a loop to read and write the contents of the file while there is something still there to process
* Make sure to close your resources!

Sample Run:

I visualize a time when we will be to robots what dogs are to humans, and I’m rooting for the machines. by Claude Shannon

The real question is, when will we draft an artificial intelligence bill of rights? What will that consist of? And who will get to decide that? by Gray Scott

By far, the greatest danger of Artificial Intelligence is that people conclude too early that they understand it. by Eliezer Yudkowsky

Did you know that social media relies on AI? Organizations utilize chatbots for clients to interact with, to post comments on social media platforms, like Twitter, and help marketers choose reliable practices to attract customers.

Submit: Submit your MySFAUsername\_textIO\_Lab.java file in the Dropbox in Brightspace by D2L.