

Name: _____

ISTE-120

Lab 13: Character IO

This lab uses a series of classes that facilitate file IO.

Exercise 1 - Copy a file (5 points)

Overview

The purpose of this exercise is to use various IO classes. To keep things “simple”, this exercise will only create one file and use only a “main” method.

Create a class called `Copy` that reads a file, copying its contents to another file. Get both filenames from the command line. Download a test file named “`test.txt`” from MyCourses to use as the source file.

Requirements:

- Use the character IO classes `FileReader` and `FileWriter` (not `Scanner` and `PrintWriter`)
- Make certain there are three command line input parameters:
 - `Input_filename`,
 - `Output_filename`, and if the output file already exists
 - `r` to replace it (overwrite it) or `a` to append to it
- Handle the `FileNotFoundException`, as well as an `IOException`
- Make sure to close both files when done making the copy
- If the input file does not exist, display a message, “Input File not found” to the screen and stop the program

Check if the output file already exists using the appropriate method in the `File` class. If the output file exists and the 3rd command line parameter is ‘`r`’, then replace the existing file. If ‘`a`’, then append the new contents to the end of the existing file. Choose the proper constructor method of the `FileWriter`. If the output file does not exist, ignore the 3rd command line parameter.

Get the input file (`test.txt`) from today’s downloads.

Write the code to read from the file whose name is the input filename (the 1st command line argument – `test.txt`). As you read from the input file, write the characters read to the file whose name is the output filename (the 2nd command line argument – use `testCopied.txt`

for this exercise). Think carefully about the condition required to set up a loop and then to stop reading when it reaches the end of file. Make sure your code handles all specific exceptions.

Sample Output

```
Command Prompt

dkpvcs> type test.txt
Test Data for ISTE-120 Lab 13
12345678901234567890123456789012345678901234567890
abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz

dkpvcs> java Copy test.txt testCopied.txt r

dkpvcs> type testCopied.txt
Test Data for ISTE-120 Lab 13
12345678901234567890123456789012345678901234567890
abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz

dkpvcs> java Copy test.txt testCopied.txt a

dkpvcs> type testCopied.txt
Test Data for ISTE-120 Lab 13
12345678901234567890123456789012345678901234567890
abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz
Test Data for ISTE-120 Lab 13
12345678901234567890123456789012345678901234567890
abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz

dkpvcs> _
```

Submit your .java files to the Lab13 Assignment folder when Exercise 1 is working correctly.

Exercise 2 – Copy a file (5 points)

Copy the Exercise 1 program (Copy.java) to another file named CopyA.java. In this class, repeat the Exercise 1 using the Scanner and PrintWriter classes to read and write. Set up the loop to read from the file whose name is the input filename (the 1st command line argument - use test.txt). As you read from the input file, write the characters read to the file whose name is the output filename (the 2nd command line argument - use testCopyA.txt for this exercise). Read one line at a time and print the line to the output file. Think about what condition to use for the loop.

Submit your .java files to the Lab13 Assignment folder when Exercise 2 is working correctly.