# Lab IO4 Buffered IO

## Objectives

In this lab, you will implement a BufferedReader and BufferedWriter to perform the same file copy that you did in the previous labs.

## Setup

1. Create a new Java project and copy the same SampleText.txt file that you have been using in the previous labs into the root directory of the project.
2. Create a package named iolab
3. Create a class named “LineCopy” with a main method

## Implementing the Code

### Step 1: Creating the IO objects

1. The buffered objects add the buffering functionality to standard FileReader and FileWriter objects.
2. You also need a String variable to hold the lines that are read in.
3. Text

   Description automatically generatedYou need to define these as follows
4. In order to create the buffered object, they have to be wrapped as follows. This has the effect of creating the reader and writer objects and then adding a buffer to each.

Text

Description automatically generated

1. The main body loop just reads a line from the input file and writes it to the output file.
2. Since the input method strips off the EOL character, you have to add it back when you write the file. If you don’t (and you can try this) the output file will just be one long string of characters.

Text

Description automatically generated

1. Text

   Description automatically generatedOnce the file has been copied, you have to ensure the contents of the buffer are flushed, or written out to the actual file. You can experiment with leaving out the flush() method and verify that the resulting output file is empty.

The final lab should look like this.

Text

Description automatically generated