# Distributed Systems – Labs

## *Week 3 - Extra*

## *File Handling in Java*

### Optional: Revision of File handling – You may skip it if familiar with

#### Writing to Character Files

5. Extract `T5\FileTest1.java`. This program uses a character oriented stream to write simple lines of text to a file. Look up the PrintWriter class in the Java API docs to see what other methods are available. Modify your code to use some of these methods.

6. Extract T6\`FileTest2`.java. This program creates a character oriented reader from a byte oriented stream to read from standard input. It also creates a character oriented writer to write data to a file. Observe the use of various methods.

**Redirection**
The standard input stream *System.in* is associated with the keyboard, whereas the standard output stream *System.out* is associated with the VDU.
Use '<' to specify the new source of input.
Use '>' to specify the new output destination.

E.g.
```
> java ReadData < payroll.dat
```

Program *ReadData* begins execution as normal. But, wheneverit executes readLine statement, it will now take its input from the next available line of text in file *payroll.dat.*

```
>java WriteData > results.dat
```

Program *WriteData* directs the output of any print and printl statement to file *results.dat*.

7. Extract `T7\FileTest3.java`. This program reads from the file created in step 2 above. Observe how string data is converted to integer data.

The `T8\Writer.java` prompts the user for input, and then writes the inputted data to a file.

The `T8\Reader.java` reads the contents of the file, and displays them.

**Command Line Arguments**

The *java* command allows us to supply values in addition to the name of the program to be executed. These values are called **command line parameters** and are values that the program may make use of.
Such values are received by method *main* as an array of *Strings.* If this argument is called *arg*, then the elements may be referred to as *arg[0], arg[1], arg[2*]….

(Java program called Copy.class copies the contents of one file to another)
java Copy source.dat dest.dat

8. Extract `T9\Copy.java`. Note how this program allows a named file to be copied to another file using character oriented readers and writers.

9. Extract T9\`FileMethods.java`. Look up the File class in the Java API docs. Looking through the code, observe how the various methods can be used to traverse, analyse and modify the file system.

**References**

Most of today's lab came from Chapter 4 of Introduction to Network Programming in Java by Graba.

https://www.oracle.com/technetwork/java/seccodeguide-139067.html#8