CSCU9T4 Practical (12th February) Spring 2018
Java Strings and File handling

In general, you should attempt the practical work here *before* the session, so you can make best use of discussion time and demonstrator assistance.

1. Register your attendance. Copy the folder **Groups on Wide\CSCU9T4\Java\Practical4** into your **T4Pracs** folder. (if it's working…)

   Open the file named **input.txt** have a look at the content. You will see that it contains a list of names and dates of birth, which are in lowercase and without formatting. For example:

   ```
   allison wesley 28011990
   peter smith 05071992
   ```

   Your task this week is to implement a command line program in Java that takes an input file with several lines each following the format described above, and produces an output file that formats all the names with either the correct (Title) case (i.e Allison Wesley) or all UPPER case (i.e. ALLISON WESLEY). The date should also be formatted as 'dd/mm/yyyy'.

   The program should take the following command line arguments:
   - An optional -u flag to indicate all upper case instead of Title case
   - The input filename
   - The output filename

   For example, running
   ```
    Java FormatNames input.txt formatted.txt
   ```
   takes the lines from **input.txt**, formats the data using Title case and places the results in the file **formatted.txt.**
   (In BlueJ this is equivalent to typing the arguments when you invoke a call to main to run the code. Strings need to be enclosed in quotes.)

   The first two lines of **formatted.txt** should look like:
   ```
    Allison Wesley     28/01/1990
    Peter Smith        05/07/1992
   ```

   If the –u flag is indicated then running the program looks like
   ```
    Java FormatNames –u input.txt formattedu.txt
   ```
   This will takes the lines from **input.txt**, formats the data using UPPER case and places the results in the file **formatted.txt**. For example, the first two lines of **formatted.txt** should look like:
   ```
    ALLISON WESLEY     28/01/1990
    PETER SMITH        05/07/1992
   ```

2. This time, you ask the user for the file name for input and for output. This is straightforward, using System.out.println. But what if the file for input does not exist, or if the file for output cannot be opened? The code below is supplied for reading and for writing…

```
System.out.println("supply filename for input:");
try {
   inputFileName= in.nextLine();
   File inputFile = new File(inputFileName) ;
   Scanner inFile = new Scanner(inputFile);
   } catch (IOException e) {
            System.err.println("IOException: " +
e.getMessage() + "not found");
   }

System.out.println("supply filename for input:");
try {
            outputFile = new PrintWriter(filename);
     } catch (FileNotFoundException e) {
            System.err.println("FileNotFoundException: " +
e.getMessage() + " not openable");
            System.exit(0);
       }
```

Adapt this code so that the user is asked for a file name until she supplies one that works, both for input and output.

Checkpoint

Demonstrate to a tutor both your programs, working with and without the -u flag on the given input file, and working with and without valid file names. If you don't finish this work today, finish it off in your own time and get it checked at the next class.

THE DEADLINE FOR CHECKING THIS WORK IS

THE LAB ON MONDAY 26th FEBRUARY 2018.