File RW - Challenge 2 - CSV Video Request List

Perhaps going more complex then we need to at this stage of the course, but if you want an extra challenge to keep you busy then by all means have a go.

The file reading and writing objects and methods we have looked at so far can process any text based file, not just simple .txt files. This includes .csv files which can be exported by any spreadsheet program (Excel etc)

It converts the data into a comma separated list which can then be read easily both by spreadsheet programs, but also any simple text processor. (Try opening a csv in Notepad for example)

Knowing this, it is easy to extend our file processing from this week's content to read and process .csv files so why not try the "real world problem" below and see how you get on.

```
A program is required to process the data relating to students who sat an exam, some of whom
did that exam at home, and identify a sample of them to request that they submit a follow up
video recording.

Write a program which will read a preformatted csv file containing the class list (who submit
ted) and an indicator of who was in lab (therefore don't have a video)  / did an email submis
sion after the official end time (thus volunteering their video).
Exclude in lab people - they have no video available
Identify 2 types of other people from the overall list:
1. email submission people - those volunteering their video
2. a specified target number randomly selected from the remainder of people


Write out a list so it can be referred to later and the candidates can be contacted to send t
heir video
```

An anonymised sample file for a small number of students is provided but assume there will be more students on an actual input file.

To simplify the problem all commas from the actual data have been removed from the file and replaced with spaces, which should make it easier to process with techniques we already know/have seen. (e.g. "surname, first-name" changed to "surname first-name")

Remember the third column, the indicator column, may have a keyword in it, or it may be empty. (would probably be simpler if the data was completely uniform, but this is a challenge)

The first row of the csv file is a header title for each column and is not strictly part of the data.

As an extra challenge could you make it so that your program first checks this header row when it opens the target file, checks that it is the right length and contains the expected heading titles. Then if it does, proceeds with the rest of the task as specified above, or if not, stops, closes the file and instead prints a warning to the screen:

```
"Sorry, unexpected file contents. Please check your input file and try again"
```

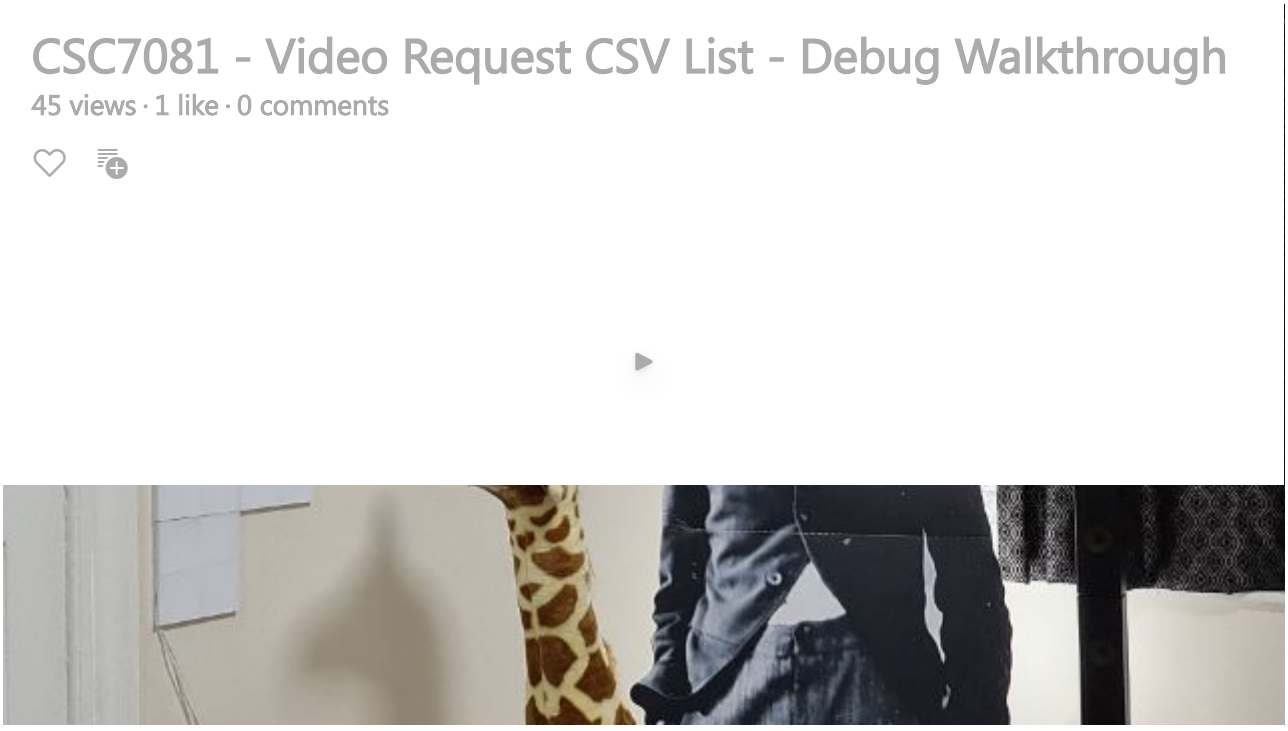
Sample file: **examSubmissionsSample.csv (https://canvas.qub.ac.uk/courses/11041/files/1291636/download?wrap=1)** **(https://canvas.qub.ac.uk/courses/11041/files/1291636/download?download_frd=1)**

A worked solution will be uploaded soon, but try it yourself without one first.

Sample solution (quite detailed, if convoluted in places. A good one to examine with debug tools and breakpoints): **VideoRequestListSelector.java (https://canvas.qub.ac.uk/courses/11041/files/1327445/download?wrap=1)** **(https://canvas.qub.ac.uk/courses/11041/files/1327445/download?download_frd=1)**

Debug Walkthrough video

Using Debug tools to analyse and study solution.



Direct Link if required: **Link (https://web.microsoftstream.com/video/189fe071-d280-495c-a2c7-81dc889edb06)**