EECS 293

Software Craftsmanship

2013 Fall Semester

Programming Assignment 10

Due at your recitation session on November 11-15, 2013

Reading

Read Chapter 8, singleton pattern (page 151 in Section 6.3), and try-finallly (page 404 in Section 17.3) in the textbook.

Programming

An old database needs to be translated for use with modern storage. In the old database, data is represented textually and fields are neatly aligned with spaces, as in:

```
1234567  MCKINLEY  NATHAN AWESOME  nmc   F13  S15
0987654  MULCAHY   BRENDAN SUPER   bm    F13  S14
0123456  LONG      STUART          slong F13  S14
2345678  GUTMAN    CAMERON         camg  F13  S14
3456789  SCHWARZ   JOSH            jschw F13  S15
```

In this example, there is only one space between AWESOME and nmc but several between JOSH and jschw. Similarly, there are only two spaces between MCKINLEY and NATHAN, but several between LONG and STUART. In general, the number of spaces is variable so as to align nicely the table. However, one space also separates first names from an optional middle name.

The output needs to remove the fifth field (which is always F13), transform the table into comma-separated values (csv), and also capitalize all names as best as possible:

```
1234567,McKinley,Nathan,Awesome,nmc,S15
0987654,Mulcahy,Brendan,Super,bm,S14
0123456,Long,Stuart,,slong,S14
2345678,Gutman,Cameron,,camg,S14
3456789,Schwarz,Josh,,jschw,S15
```

Create a repository called csv.git where you will post your submission.

For testing purposes:

- Submit exhaustive JUnit test cases, and

- Prepare a separate class with a main that reads a space-formatted table from standard input and outputs its representation as csv to standard output.

Submit a separate text file to document the error handling architecture that you will follow in your implementation. The architecture document should describe your error handling choices such as a strategy for handling erroneous user input, decisions on local or global error handling, error propagation through the code, presence and location of a barricade, and the other factors in the defensive programming checklist at the end of chapter 8.

Discussion Guidelines

The project discussion will focus on defensive programming.