Cemetery Lab

Disclaimer: this lab involves a database from a real cemetery. If you are sensitive to issues of mortality, grief, and loss, please see me about an alternative assignment.

Also, this assignment is intended to simulate a very real industry project. The client has a need, and you have the tools and creativity to create a program that satisfies their need. To better simulate a real-world programming situation, there is a hard deadline for when the project needs to be completed by. Assume that you will only get paid (by means of a grade) if the program satisfies the requirements and is delivered on time. Your deadline time is 5 class periods (2 weeks). You may work within your Capstone team (a team of 2-3), work solo or work with a different team of 2-3.

In this lab you will find the full database of the residents of the St Mary Magdalene Cemetery in London in **`cemetery_orig.txt`**. Each burial site has information on the deceased's name, burial date, age-at-death and address. The client (your teacher) needs this database to be read into some kind of data structure and searchable with the following features:

- Search by last name – the user types in a last name and all residents with that last name are shown with their full information.
- Search by street name – the user types in a street name and all residents that lived on that street are shown with their name, burial-date, and age at death. Or give the client a list of streets to pick from, as determined by when the database is read in.
- Show residents sorted by name, burial date, age at death or address.
- Show interesting data: average age at death, the year with the highest mortality rate, the oldest age at death, the street with the highest mortality rate.

This lab is completely open-ended with regards to coding choices for algorithms, data structures, programming language, interfaces (text/GUI), and coding strategies.

Items to note:

- The data in the file is not consistent: most ages are listed as decimal numbers, but some have a number followed by 'd', 'w' or 'm' (# of days, weeks, or months old). Depressing.
- The addresses are not consistent: some just have the street name, some have the house number and a street name, some have a street name followed by a postfix term (like MDX).
- Some cemetery residents have the same name (and some also with the same address).

Items to consider:

- What kind of data structure(s) would be best for ease of coding and efficiency?
- Is object orientation (Person object) a helpful design choice here?
- On the data-file inconsistency, do you take the time to scrub the data file, or do you write the code to be able to read in and correctly process the variations in data?
- Since some addresses are more complex and detailed than others, what part of a complex address seems like the core part of a street name for searching?
- How can you make your program more user friendly?
    - There are a finite number of unique core street names. Will you make the client type in a street name, or select one from a list or drop-down menu?