

## Project 2, CISC181 Fall 06

**Due Sunday Nov 19 midnight, paper Monday in class**

**Domain choice due November 8th**

### Assignment

Please read the online FAQ for this project, which I will add to as I receive questions.

Write a simple data management system. The example we discussed in class was for inventory management for a pet store, but you must come up with your own domain.

Your domain must have different categories of things to keep track of. In the pet store example, we discussed multiple inventory lists: pets, food, toys, treats, accessories.

For each category your program will be able to **find** items by name or ID number; **change** information in an item; **delete** an item; and add items. You will be able to **sort** any of the categories by name, ID, or price.

### Structure

Your final implementation will have an array of linked lists, where each list is a different “type” of data for your domain.

To simplify your system, have a single struct or class to hold data. You may name the structure according to your domain; for the pet store, I might call my structure “item”. Your structure may have some fields that are not used by every category - in that case, be sure they have been set to appropriate values that indicate they are not valid.

This “item” structure is **not** the same as the Node that your linked list is composed of. Instead, each Node of the list will have a pointer to an instance of this structure. This is so that your list code does not need to be modified.

### Domain

Select your domain. No two students may use the same domain. All domains must be approved by the instructor. Email me your chosen domain with example information and categories by November 8th. When two students accidentally

choose the same or similar domains, then I will select the first sufficiently specified application, and the other student will select another domain.

Your domain must have different categories of things to keep track of. In the pet store example, we discussed multiple inventory lists: pets, food, toys, treats, accessories. For each category items should have multiple attributes, and all categories must at least have name and ID number attributes.

You will need to acquire data for your domain. Attribute any sources you use (websites, textbooks, etc.).

## **Data**

Put the data in a comma-delimited file (.csv from a spreadsheet is fine). Read the data only once, allow the user to update and modify it, then at the end of your program write the changed data to a different file.

Each line of the data file should represent one item. The first column will specify the category. Items may appear in a data file in any order; your program may not assume they are ordered in any way.

## **Menus**

You will have a multi-level menu system of numbered choices, as discussed in class. The user can choose to exit the program from any point in the menu, at which point the program will return to main, write any changed data to a file, and print a message before exiting.

## **Testing**

Write testing code that will test the functions in your code. You should have a separate testing suite for your linked list class, and another for your project-specific code. Your makefile should run the testing programs when you change your code. This testing is to test your code's correctness - be sure it is complete and tests boundary cases (ends, beginnings, middles, high and low values, etc.) This is **not** related to the demonstration of your project which you will do in your scripting.

## Grading

percent	
60	Code: modularity, encapsulation, and decomposition; clarity of design and intent; productive comments; functional and efficient code.
20	Menus: user-friendly, clear and tidy, functional
20	Testing functions, makefile