

Objective: Become more familiar with dynamically allocated singly linked lists and the basics of software design. Recall from CS 120 that top-down design is a problem-solving method in which the programmer breaks a problem up into its major subproblems and then solves the subproblems using techniques such as stepwise-refinement to derive the solution to the original problem. All subproblems should translate directly into functions used by the final program.

Program Description: The program will read in a file containing commands that modify list of animals. Animals will be *added* or *deleted* from the list as specified in the file (one command per line). There is also a command to display the list at multiple locations in the file.

Requirements: Your program (modular) must perform the following operations:

- Read the animal file (build list).
- Print the animal list.
- Display the number of each type of animal in the list.
- Modify/Display the list after a specified animal is added/deleted.
- Display the number of each type of animal remaining in the list.

Sample:

```
pets = ["cat", "dog", "bird", "cat", "bird", "dog", "cat", "cat"]
Print( pets )
[cat, dog, bird, cat, bird, dog, cat, cat]
Print( count( pets, "cat" ) )
4
```

Deliverables:

- Program—fully documented.
- A program design sheet. Describe all classes and methods needed to implement your program.
- Programming Log:
 - Record of things you encountered/learned while implementing your program.
 - Record the time required to design and implement your program.
- Output—proof that your program worked. Two data files have been posted on the class website.

If you have any questions regarding this assignment, do not hesitate to contact me. Start working on this assignment as soon as possible.