

CS163 Lab Session #2 – Abstract Data Types

Please complete this to become familiar with ADTs.

Submit code to the CS199 D2L dropbox. Limit the time invested to 1 hour and 50 minutes maximum.

Coding: *With this lab, we will be working with an existing class implementing a list ADT for a journal. The data structure is a linear linked list of journal entries. You have access to the .h files to examine the class interface and member functions on D2L's online "locker". Your job will be to implement functions to experience manipulating the lists in unix.*

Develop an ADT: *In literature classes you may have learned about journals. Keeping a journal when programming allows us to begin to learn from our mistakes and keep a log of ideas for improving our programming skills in design, syntax, and debugging. As such you will be implementing the code for a Journal ADT for this exercise.*

____ Step 1. Examine the list.h file and the journal .h file and answer these questions on your own. If you have questions with any of these, post them on D2L, the Scribblar link for Lab #2 or contact karlaf@cs.pdx.edu:

- a. Why is "int" returned from the functions_____
- b. Examine the arguments for the add functions. Where is the information read in from the user? _____
- c. Examine the retrieve (or find) functions. How do you think the functions communicate back to the calling routine the information found?
- d. What should the destructor for the list call do?

____ Step 2. Begin implementing the member functions, in a .cpp file and upload these to D2L's CS199 dropbox:

- a. Constructor list();
- b. Destructor ~list();
- c. The add function int add(journal & to_add);
- d. The find function int find(char title[], journal & found);
- e. The display function int display();

____ Step 3. Compile and run:

- a. Modify main to call your functions
- b. Download the files from D2L's online locker for this lab
- c. Compile: g++ *.cpp *.o
- d. Run: ./a.out

_____Step 4. **Develop the test plan:** *For each member function that you plan to write, think about how to test it – what flow of control exists in the member function and how would you test out all conditions:*

Test Case(s)	Expected Result	Verified? (yes/no)
Create a journal object and destroy it		
Add 1 entry and display that entry		
Try to display entries when nothing exists		

Verify correctness: Using the above test plan, create a test program that tests the interactions of all functions together.

_____Step 5. **Challenge (*optional*):** Implement the code to remove the last item from a linear linked list recursively. Submit your code to D2L's CS199 dropbox.

Self-Assessment: *What could you do to improve for next time?*