CS163 Lab Session #2 – Building an ADT

Process:

- Each lab question is designed to be done in a step by step fashion. The success of later steps depends on earlier. For each step, discuss your findings with your team member before moving forward.
- **Begin with the pre-lab questions. For inclass students, these solutions are vital to the code that will be written during lab time.** It will also provide an avenue for receiving feedback before working on individual programming assignments.
- **Only after developing the pre-lab solutions should you individually work on the coding questions in the lab assignments.**
- **Testing is a vital part of software design. Most labs will require test plans be developed. Then, after each function is written**, test it out – following the test plan developed during the planning process

Goals:

- Strengthen knowledge and gain experience planning, developing, and evaluating C++ classes and member functions.
- Learn how to apply classes to build ADTs
- Experience implementing portions of linked list software.
- Experience critical thinking through collaboration.
- Build test plans and experience unit testing as member functions are developed

Syntax Reminders:

- All code development must be done on unix using vi, vim, emacs, pico or nano
- You will be writing code using 3 different files for all 5 programs this term:
    a. The header file (.h) should contain the #includes, prototypes, structures, and class interface. Don't implement any functions in the .h file
    b. The class implementation file (.cpp) should #include the .h file (#include "journal.h"), and include the bodies of all member functions. Write the constructors and destructors first:

        ```
        Journal::Journal() { /*body*/ }
        ```

    c. The test program's implementation file (.cpp) should also #include the .h file that you create and then call the member functions.
- To compile on unix type:

    g++ journal.cpp main.cpp /*using your file names of course */