

Programming Assignment #5

CS 163 Data Structures

Submit your assignment to the **D2L Dropbox** (sign on via d2l.pdx.edu)

Programming - Goal and Data Structures: The goal of this program is to create a graph abstraction using an adjacency list (an array of vertices where each element has a vertex and a head pointer to a LLL of edges for adjacent vertices). We will be implementing only the following algorithms (e.g., member functions):

- a) **Build an adjacency list** (An array of vertices, where each has a head pointer to a LLL of nodes. The nodes indicate which vertices are adjacent.)
- b) The graph will be a **weighted graph**, which means there will be data with each edge (i.e., in each edge list's node).
- c) **Insert** a connection between two vertices (i.e., inserting a node into the edge list)
- d) **Traverse** the adjacency list, displaying which vertices are adjacent. And, display all.
- e) **EXTRA CREDIT** for an implementation of a **depth first algorithm**
- f) Destroy, deallocating all dynamic memory (e.g., destructor)
- g) *** THERE IS NO REQUIREMENT FOR INDIVIDUAL DELETE FUNCTION!

Application: As we go through a day we encounter many decisions. Should I leave the house at 6:30? If I do I avoid rush hour traffic, get into work early, and am more productive. If I wait and leave the house at 8am, then I am in the worst of rush hour traffic, get to work an hour later, and am already tired and am not ready to work. These both lead to the next level of decisions that I am faced with. If I get in early, I end up having time for a nice long lunch. If I get in late, I am stuck with a fast lunch or no lunch at all.

At lunch, I have the option of eating a hearty lunch with many fruits and vegetables, I can eat a fast food burger and fries, or I can drink a protein shake. If I choose the hearty lunch, I will add 5 years onto my life span (ok, just kidding). If I eat the fast food burger (and fries!), I will gain weight. If I drink the protein shake, I will soon be grumpy due to low blood sugar. Hopefully, you get the idea. With each decision there are resulting ramifications. Then we make the next decision. And so on.

Your job for this assignment is to allow the client to enter in at most 10 different decisions (vertices) that they can make. Based on these there are different outcomes (edges) that end up leading to different decisions.

Your primary job in this assignment is to experience creating your own adjacency list based on the decisions and outcomes specified by the client program. The adjacency list will be an array of vertex objects and a head pointer for each linear linked list representing the edge list. **Create the code to allocate an “adjacency list” for a graph. The adjacency list should contain:**

- (1) Vertex Information – A decision
- (2) Head pointer (to an Edge List)
- (3) Visit indicator (optional – only needed for extra credit)
- (4) Edge node – Contains the information about one of the results of making this decision. A decision that results in three different possible outcomes will have three edges in its edge list!

Things you should know...as part of your program:

- 1) Do not use statically allocated arrays in your classes or structures. All memory must be dynamically allocated and kept to a minimum!
- 2) All data members in a class must be private
- 3) Never perform input operations from your ADT in CS163
- 4) None of your public member functions should have “node” data types as arguments. However, you SHOULD have private RECURSIVE member functions that do take node pointers as arguments
- 5) Global variables are not allowed in CS163 – not even in your main
- 5) **Do not use the String class – not even in your test suite! (use arrays of characters instead!)**
- 6) Use modular design, separating the .h files from the .cpp files.
- 7) Use the iostream library for all I/O; do not use stdio.h.
- 8) Make sure to define a constructor and destructor for your class. Your destructor must deallocate all dynamically allocated memory.
- 9) Remember that 20% of each program's grade is based on a written discussion of the design. *Take a look at the style sheet which gives instruction on the topics that your write-up needs to cover.*