**Anthony Cormican – Dijkstra Assignment Week 3**

In approaching this program, I decided on using an edge-list implementation, as it was better suited to a linked list data type. By using a series of linked lists to represent the graph’s nodes and edges, I was able to use pointers to node objects instead of just having a ‘conceptual’ model. In my program, each node is a real object in memory. The graph class is generic and can operate on any node data type. The edge list approach also easily facilitated the use of recursion to parse the graph structure efficiently.

This assignment helped to clarify the following concepts for me:

* STL iterators (used with dynamic vectors)
* dynamic data structures and memory management
* using template classes to provide greater class abstraction
* passing by reference
* the innate recursive qualities of the graph/linked link structure.

Overall, I found that this program required careful thought about how the basic ADTs were to be structured. Each ADT is built on top of the previous and small errors in logic quickly become hard to debug. My goal was to create a program that could accommodate graphs of any type and size and create and release memory correctly. I have managed to make this happen!