CS 2420  
Graphs

# Purpose

# to familiarize you with graphs

# to help you understand depth-first search

# to aid in understanding shortest-path algorithms

# Tasks

* Create a Graph Class
  + This class is similar to the one in the book, however in this graph you need to use weighted edges.
    - You do not need to use templates
  + Include
    - Constructor ()
    - Destructor()
    - add\_edge()
    - add\_vertex()
    - is\_edge ()
    - Depth-First search ()
      * This function is here to help you understand this concept.
      * It is somewhat separate from the rest of the project.
      * Start with the first node and simply print the node as you reach it.
  + You may use the code from pages 748-752 to start with
* Use any of your previous classes to be used as a **SET** class.
  + The set class should wrap around your other class.
    - In other words the other class is a data member of your set class.
  + You should not allow duplicates in your set
  + It should provide the following functions.
    - Insert
    - Is\_in
* Implement Dijkstra’s Shortest-Distance Algorithm
  + Pseudo-code is on page 772
* Create a Driver
  + Create a **non-interactive** driver
    - No menu
    - Use the sample graph that is in the book on page 766.
    - Initialize your graph using
      * add\_edge and add\_Vertex
    - Do a depth-first search and display the order of the vertices traversed
      * Display the order in which dfs is executed.
      * Use pg 754
    - Perform Dijkstra’s Algorithm on all vertices in the graph
      * Display a vertex pair and the shortest path cost.

Turn in:  
Code (.h and .cpp files),Windows executable