We were asked to create a graph of n Nodes for a given density. The edge costs were generated randomly given the maximum edge cost. The goal is to find the shortest path from Node s to Node d using Dijkstra’s shortest path algorithm. The program is broken into various classes. Here is a description of all the involved classes:

1. **Graph :** This class provides a constructor to create a random graph given some density, maximum edge cost, and total number of nodes. The graph is represented using a 2-d vector and this class provides a few methods to edit the generated graph. However, those methods are not required for this assignment.
2. **PriorityQueue:** This class implements a generic min heap data structure which is needed for Dijkstra’s algorithm to fetch the next node in pursuit of the shortest path using greedy approach.
3. **NodeInfo:** This class is an ADT to maintain two pieces of information for every node: What is each node’s parent while traversing through using Dijkstra’s algorithm and what’s the cost associated with the getting to the node from source node. Some operators like “<”, “>”, “==” are overloaded for this class as the priority queue is implemented for this data type in the algorithm
4. **ShortestPath:** This is the class that implements the mechanics of the algorithm and has an object of Graph and PriorityQueue data type.

This assignment has helped me learn basics of Graph theory as my background is not computer science. It has also enabled me to be able program multilple classes in C++ and how should they interact.