Final Project

Omer Bareket

208548834

Explanation:

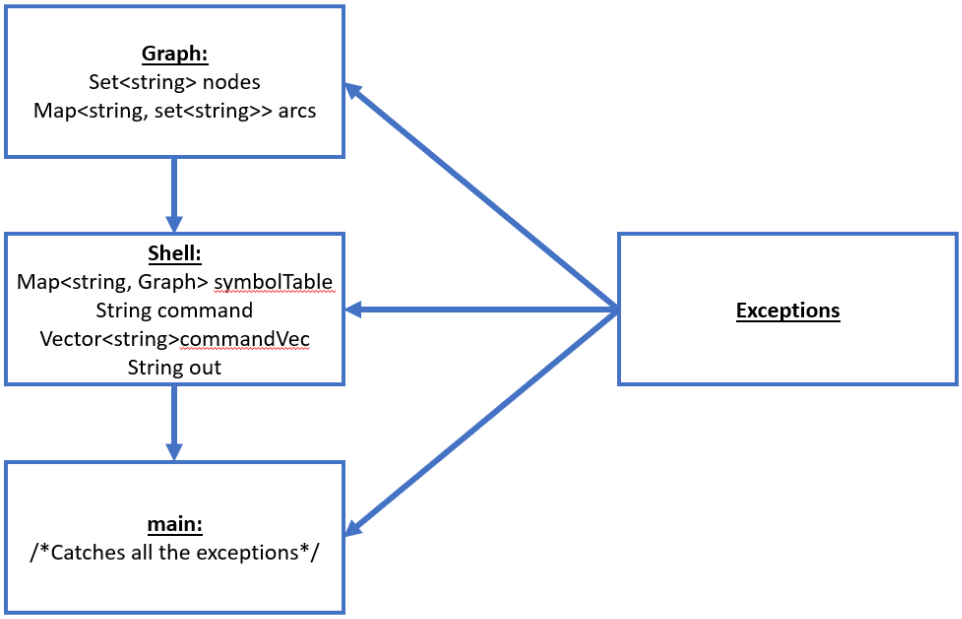
The program has three 2 main classes: Graph and Shell.

The first thing the program does is that it figures out whether it is working in automatic mode or interactive mode using argc so that it knows whether to write the output to cout or to the output file. The Graph Class is a class the is the basis for every graph. Inside it are a set of string which are all the node names, and a map<string,<set<string>> in which the key is the name of source node, and the values is a set with all the nodes that node has an arc to. The shell class is where all the computation happens. It is comprised of a map<string, Graph> which houses all the graphs we have in the system (symbolTable), input string (command), vector to parse command (commandVec), and output string (out).

First the shell gets is a string which is the input. The first thing is does is "clean" the string of spaces and black characters. If it finds an error in the input in this stage it throws an error. After we have a clean string, it figures out whether this is an '=' operation.

It sends the terms to a recursive function that breaks the function down until it is left with only a graph literal or a graph name. It then checks whether that graph exists. It sends the result up in the recursive tree. The final graph after all the recursion is done is sent to the main Parse function which uses that term to do the appropriate function.

Design Chart:



Structure of class Graph

Graph:
Set<string> nodes
Map<string, set<string>> arcs

Example: g={a,b,c|<a,b>, <a,c>, <c,b>}

Set<string> nodes:

a	b	c

Map<string, set<string>> arcs:

key	value
a	{b,c}
c	{b}