Lab 04

Resources

- Discussion slides (./lab04slides.pdf) - Copy of the presentation given by the TA
- count.h (./count.h)
- count.cpp (./count.cpp)
- main.cpp (./main.cpp)

Idea

In this lab, you will learn about separate compilation and makefiles. We will use the g
you built in the first lab and break it into separate files. The focus of this lab is not on
implementation of the graph, but on how it can be broken up. However, if you used g
variables or other such things, you will have to change your code around a bit.

In-lab assignment

Description

You will modify your graph struct and the corresponding menu logic into multiple fil
These files should separate the "graph" logic from the "main" logic. You will also writ
play with a makefile to become more familiar with how to use them.

Requirements/Deliverables

The flow of this lab will be more like a tutorial than last labs. Please follow the steps a
they are written. On steps that ask a question such as what was compiled (these are
highlighted in this color), please answer them in a separate text file called answers.tx

Label the answers using the number of the question. Example (though your answers should be much more expansive):

Questions:

1. Who is the professor of this course?
2. What lab is this?
3. What is the answer to life, the universe, and everything?

answers.txt

```
1. Dr. Nemo
2. Lab 4
3. 42
```

- Change the type of your graph from a struct to a class. Be sure to properly use th public and private identifiers. In this case, all of the data members should be pri Only the functions I asked you to write in lab 2 should be public.
- Break up your class defintion so that only the declarations are inside of the class Move the function definitions outside of the class.
- Create a graph.h file that contains just the declaration of the Graph class. **No functions should be defined in this file!**
- Add in the ifndef directive and all it needs to work to prevent multiple includes f happening
- Create a graph.cpp file that contains the *definitions* of the functions in your Gra class. Don't forget to include your graph.h file. Also don't forget the Graph:: pref your functions. Note that you have already written these functions, you can just the implementations here. **This file should not contain a main function**.
- Move all of the menu logic that manipulated your Graph and your main functior a file called graphmenu.cpp. Again, you must include your graph.h file in order t your Graph class.
- When all of this is done, compile your program units separately and then link th together:

```
g++ -c graph.cpp
g++ -c graphmenu.cpp
g++ graphmenu.o graph.o -o graph
```

- Then you should run `./graph` to make sure everything is working correctly.
- *To save yourself some heartache, you will need to get the previous parts worki before you can move on to playing with the make files*
- Write a make file similar to the second (or third) one presented in the lab discus using your graph.cpp and graphmenu.cpp files.
- Remove all of your previously compiled things with:

```
rm *.o graph
```

- Run the makefile with the `make` command. Run `./graph` to ensure everything worked correctly.
- Modify the graphmenu.cpp file so that add edge is now done when the user ente and add vertex is done when the user enters 2.
- Run the `make` command again. 1. What happened? Which files were compiled? Discuss it with your neighbors/the TA and ensure you understand the reason.
- Modify the Graph implementation (the .cpp file, not the .h) by adding a print statement to each function saying the function name.
- Run the `make` command again. 2. Again, what happened? Which files were com And why?
- Now, modify the Graph class by adding a method that returns the number of ve in your current graph, so you should have the following inside your Graph decla in your .h file and the corresponding definition inside your .cpp file:

```
int num_vertices();
```

- Now recompile your program using `make` once more. 3. What happened? Which were compiled this time? Why is this different from the previous 2 examples? Di edit graphmenu.cpp? Did it recompile? Why?
- Now add a menu option to your graph that allows the user to get the number of vertices in the graph.
- Again, recompile with `make`. 4. What was compiled this time?
- Run `make clean` (you might have forgotten to put a clean in the makefile, in wh case you need to do this now). 5. What does this accomplish?
- Recompile once more with `make`. 6. What was compiled this time?

Submission

When finished, please compress your answers.txt, makefile, graph.cpp, graph.h, and graphmenu.cpp into a zip file called lab04.zip. Submit this archive to Sakai in the La assignment area.

Hints

- Don't tug on superman's cape

Grading Distribution

- 3 points for a correct makefile
- 9 points for breaking up your program correctly including:
  - 3 points for a good .h file with ifndef, class declaration, no definitions, etc.
  - 3 points for a graph.cpp file which only contains defintions of the functions graph.h file
  - 3 points for correct implentation of graphmenu.cpp and usage of the Graph
- 5 points for the answers.txt
- 3 points for good style

Optional Enhancements

You can try to make the fourth makefile work with the project. I still only want a mal comparable to the second or third one for this lab.