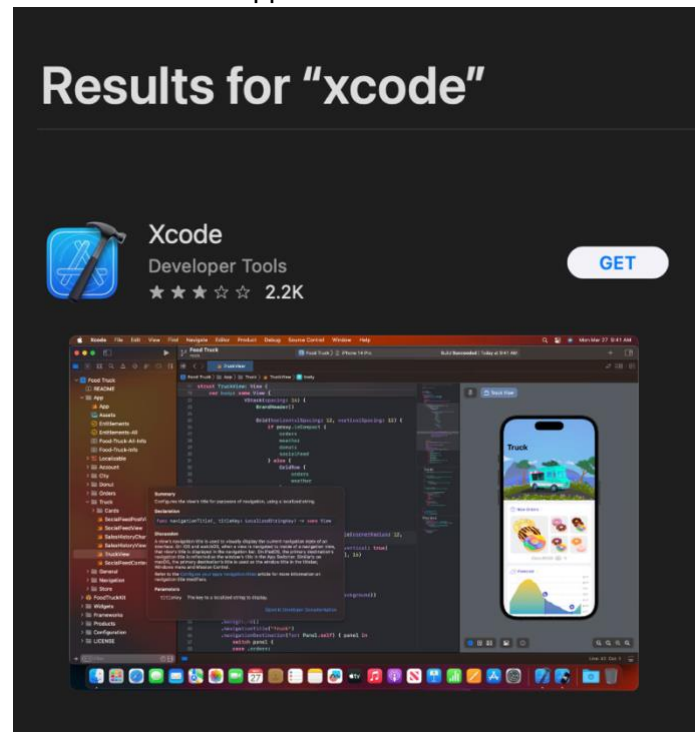## P1. Installing All Needed Software

XCode:
Note - XCode is a fairly slow download, please start with this and ensure that there is at least 20 GB of space on your system.
 1. Download the XCode IDE from the AppStore

  2. Allow XCode to install. We will configure its settings in a future step
XCode requires an updated MacOS version, make sure your system is up to date

Homebrew:
  1. Install homebrew by visiting the link: https://brew.sh
  2. copy the command line:
     /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
  3. open terminal and paste this line and press entre, it may ask for your password
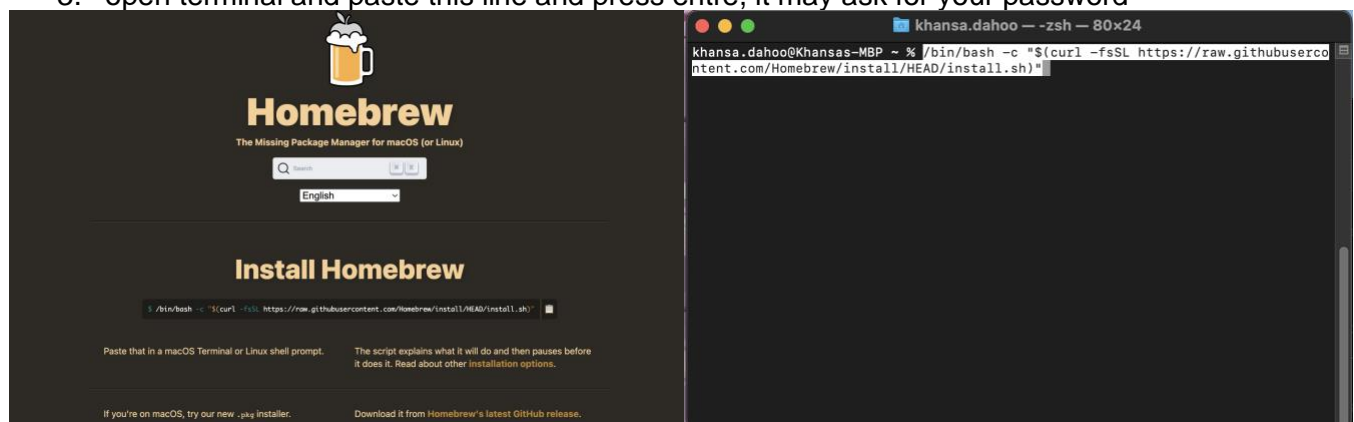
4. the output should be as follows, press entre to begin the installation

5. run the following commands to add Homebrew to your path:
   a. (echo; echo 'eval "$(/usr/local/bin/brew shellenv)"') >> /Users/[user-name]/.zprofile
   b. eval "$(/usr/local/bin/brew shellenv)"

Glew and GLFW:
   1. open terminal and paste the following commands to install glew and glfw respectively
      a. brew install glew
      b. brew install glfw3

   2. this will create a 2 files under: /usr/local/Cellar named glew and glfw
      we will be moving these folders out of the hidden directories to make it easier to access
   3. under MacintoshHD, press Command + Shift + . to show all hidden folders, click on usr
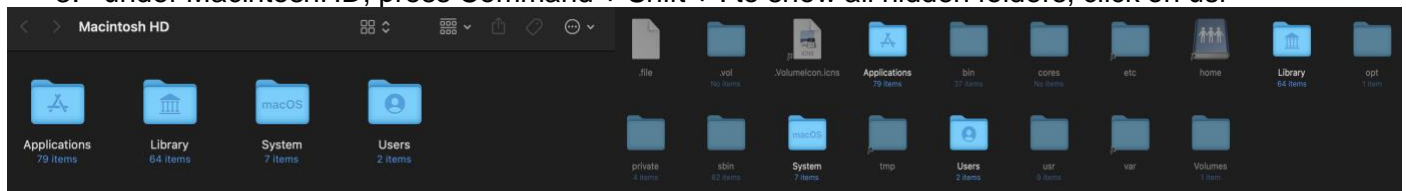
   4. navigate to: local -> Cellar and copy the glew and glfw folders
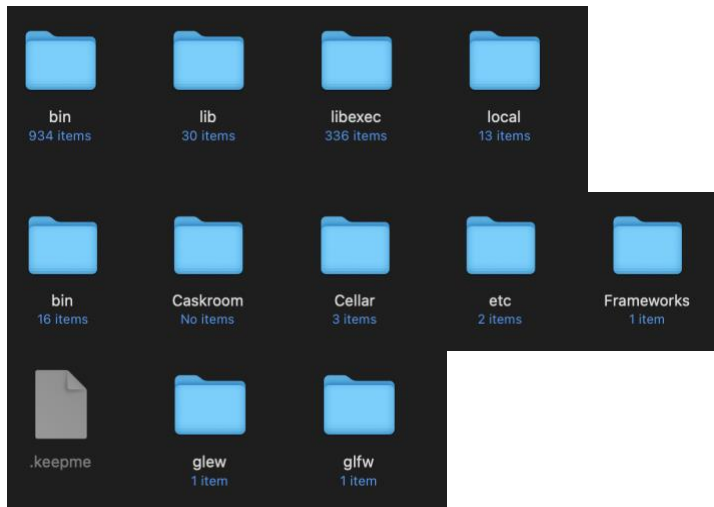
5. Return to MacintoshHD and navigate to User
6. Create a new folder named include and paste both folders within it
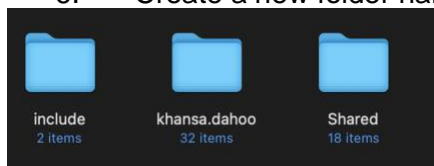
P2. Configuring XCode

1. If this is your first time using the XCode IDE, it may ask you which devices you will be developing for. Check whichever devices you are interested in, make sure macOS is selected and press install
2. Press on create a new project once XCode is finished with its initial installations

3. In the new window, press macOS and select Command Line Tool
4. Fill in the required fields with the name and organization of your choosing
    Make sure the Language is set to C++
5. Select a location to place the project folder
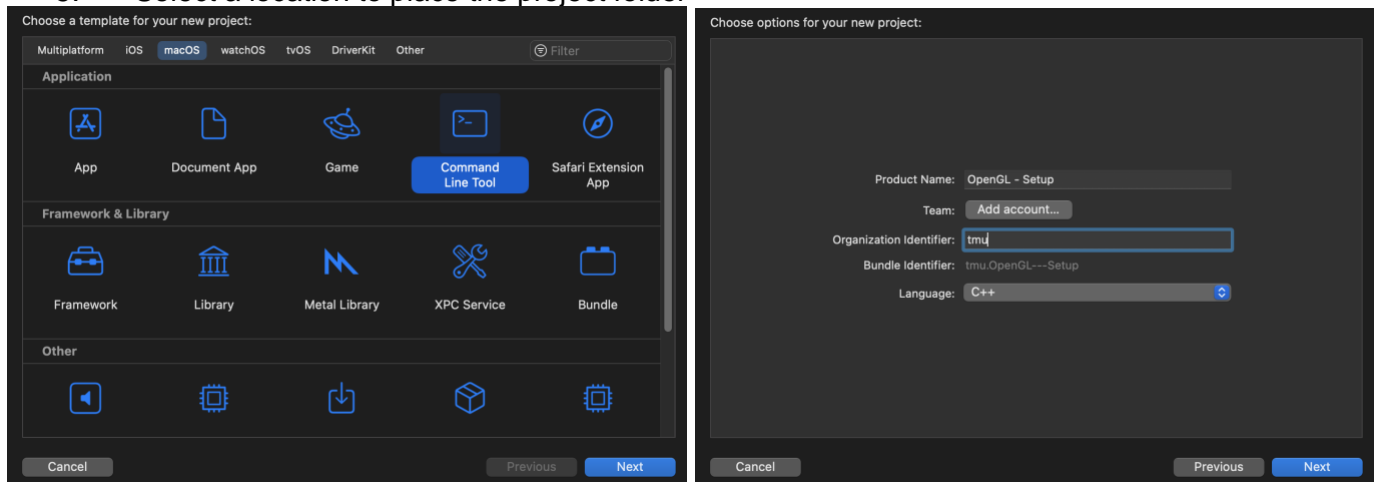
P3. Configuring Project Setting

The project should have a main c++ file which runs a simple hello world program. We must include the path to OpenGL, GLFW and GLEW in order to use all libraries for this course.

1. Navigate to the Project settings file
2. Under build settings, look for "search paths"
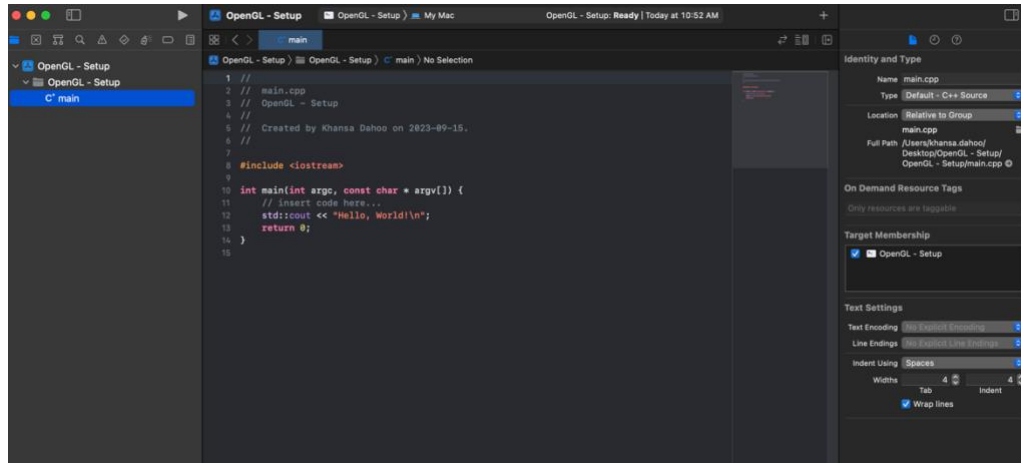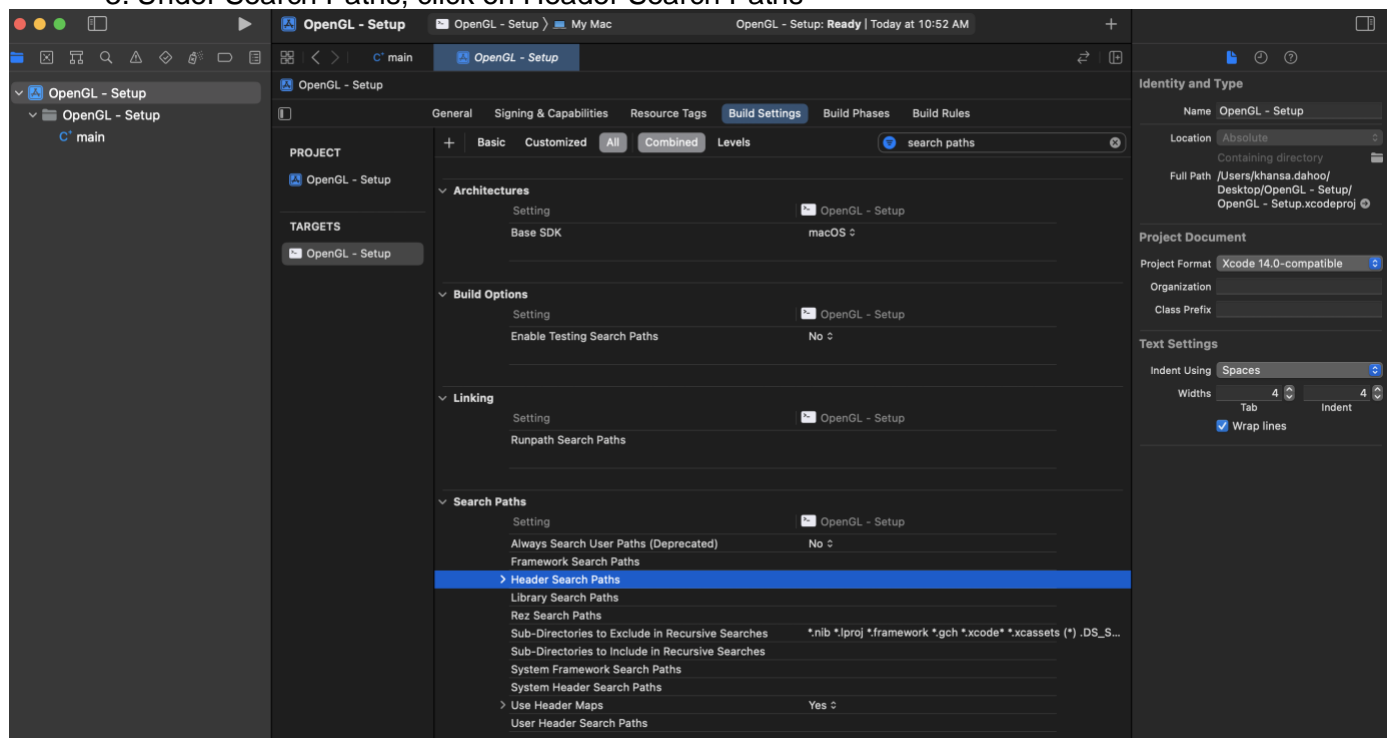   You may need to click on "All" to show the all configurable settings
3. Under Search Paths, click on Header Search Paths

4. Double Click on the right side and press the plus button to add a new path
5. Paste these paths into the field:
   a. /Users/include/glew/2.2.0_1/include
   b. /Users/include/glfw/3.3.8/include

Note – These paths can be different on your system, you can check by navigating through the GLEW and GLFW folders and checking their info
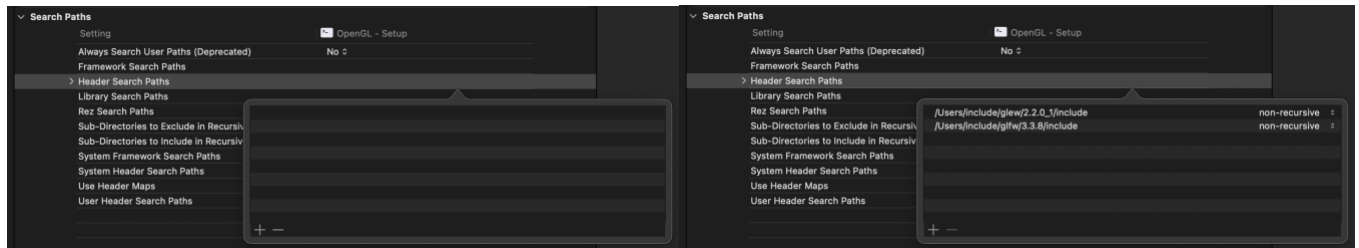
6. Double click on the right side of Library Search Paths to add new paths
7. Paste these paths into the field:
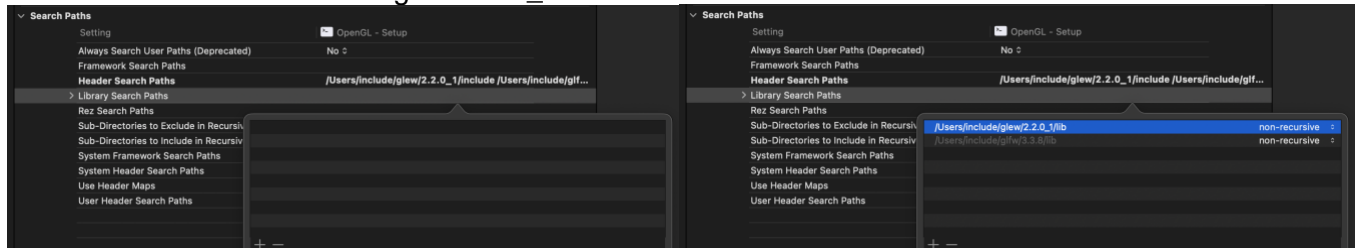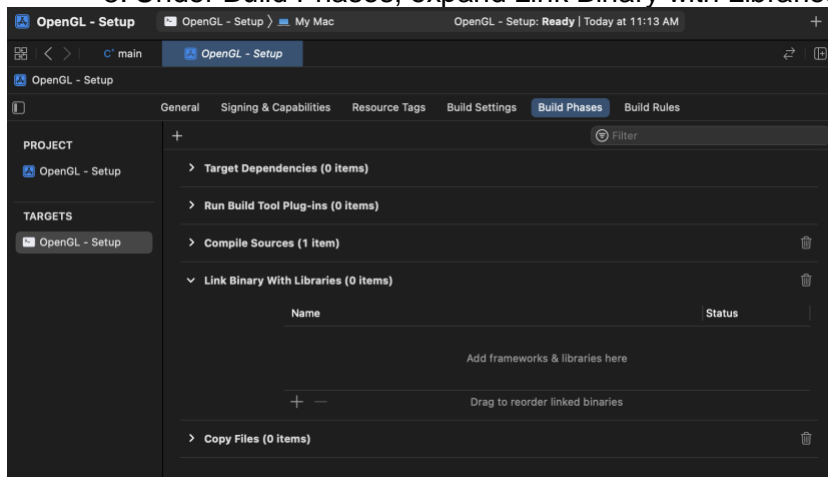   a. /Users/include/glfw/3.3.8/lib
   b. /Users/include/glew/2.2.0_1/lib

8. Under Build Phases, expand Link Binary with Libraries

9. Add a new library: opengl.framework (this should be preinstalled with XCode)
10. Add a new library: glut.framework (also included in XCode)
11. Add a new library: press add other -> add file
    a. Navigate to Macintosh HD -> Users -> include -> glfw -> 3.3.8 -> lib -> libglfw3.3.dylib
       Press open on the libglfw3.3.dylib
    b. Navigate to Machintosh -> Users -> include -> glew -> 2.2.0_1 -> lib -> libGLEW2.2.0.dylib
       Press open on the libGLEW.2.2.0.dylib
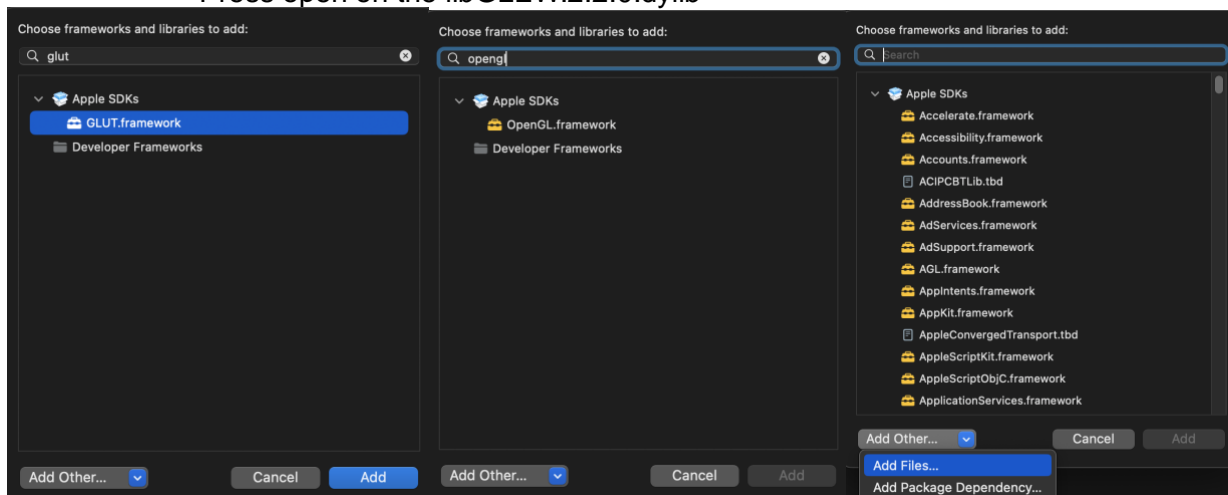
COMPLETED SETUP:

You should now be able to copy and paste any of the example codes to the main.cpp file and run them. Do not forget to replace #include <GL/freeglut.h> with #include <glut/glut.h> as freeglut is not supported on MacOS.

Some functions will throw a warning that they are depreciated, this is because Apple no longer supports OpenGL as of version 4.3. This should not affect functionality and you can mute these warnings by adding: #define GL_SILENCE_DEPRECATION to the start of your script.

An example of the final output is as follows: (using the VboDrawElements.cpp file found on D2L)