#define \_HAS\_EXCEPTIONS 0

// cl /FAs test.cpp

* Create new Visual Studio project:
  + Visual C++, Windows Desktop, Windows Desktop Wizard
  + Uncheck: Pre-compiled headers and Security Development…
  + Check: “Empty Project”
  + Go to GLFW and get the example code (Documentation, then “Getting Started”, then go to bottom “Putting it together”)
* Get glad (for OpenGL)
* Get GLFW – dlls
* Add two folders “include” and “lib”
* Under “lib”, add two more “x86” and “x64”
* $(ProjectDir)\include
* $(ProjectDir)\lib\x86 glfw3.lib
* $(ProjectDir)\lib\x64 glfw3.lib
* Note: You will have to do this for each build (Debug/Release AND x86/x64), assuming you will use them all
* Download glm
  + #include <glm/glm.hpp>
  + #include <glm/vec3.hpp> // glm::vec3
  + #include <glm/vec4.hpp> // glm::vec4
  + #include <glm/mat4x4.hpp> // glm::mat4
  + #include <glm/gtc/matrix\_transform.hpp>   
    // glm::translate, glm::rotate, glm::scale, glm::perspective
  + #include <glm/gtc/type\_ptr.hpp> // glm::value\_ptr
* Change “gladLoadGL([glfwGetProcAddress](https://www.glfw.org/docs/latest/group__context.html" \l "ga35f1837e6f666781842483937612f163));” to
  + gladLoadGL([glfwGetProcAddress](https://www.glfw.org/docs/latest/group__context.html" \l "ga35f1837e6f666781842483937612f163));
* Add the lib dependencies to glfw3.lib
* Go to the “glad” page and download the package. We need “glad.c”
* Add “glad.c” to the project
* Replacing linmath with glm
  + Comment out linmath
  + Search out the errors:
    - Replace m, p, mvp with glm::mat4
    - Add view matrix
    - Replace rotation with glm::rotate (around z)
    - Replace linmath ortho with glm::ortho
      * Point out -1 to 1 “square” (or cube)
      * Uniform Device Coordinates
    - “mvp” stands for “model view projection”
    - Note that it’s “backwards” because matrix math goes “in reverse”, so it’s:
      * mvp = p \* view \* m  
        **NOT**
      * mvp = m \* view \* p
* Replace ortho with perspective
* Add lookAt()
* Disable back face culling – but how??
* Add ability to load vertices from a file
  + Talk about c style arrays
  + …and pointers
  + Load the values into the vertex buffer



Checkpoint: add something that has at least 12 triangles, with multiple colours, that shows “something recognizable”.