Overview

In this course, many assignments are paired with tutorials to help you to learn OpenGL and build the computer graphics skills that you will need to successfully complete your coding and design work.

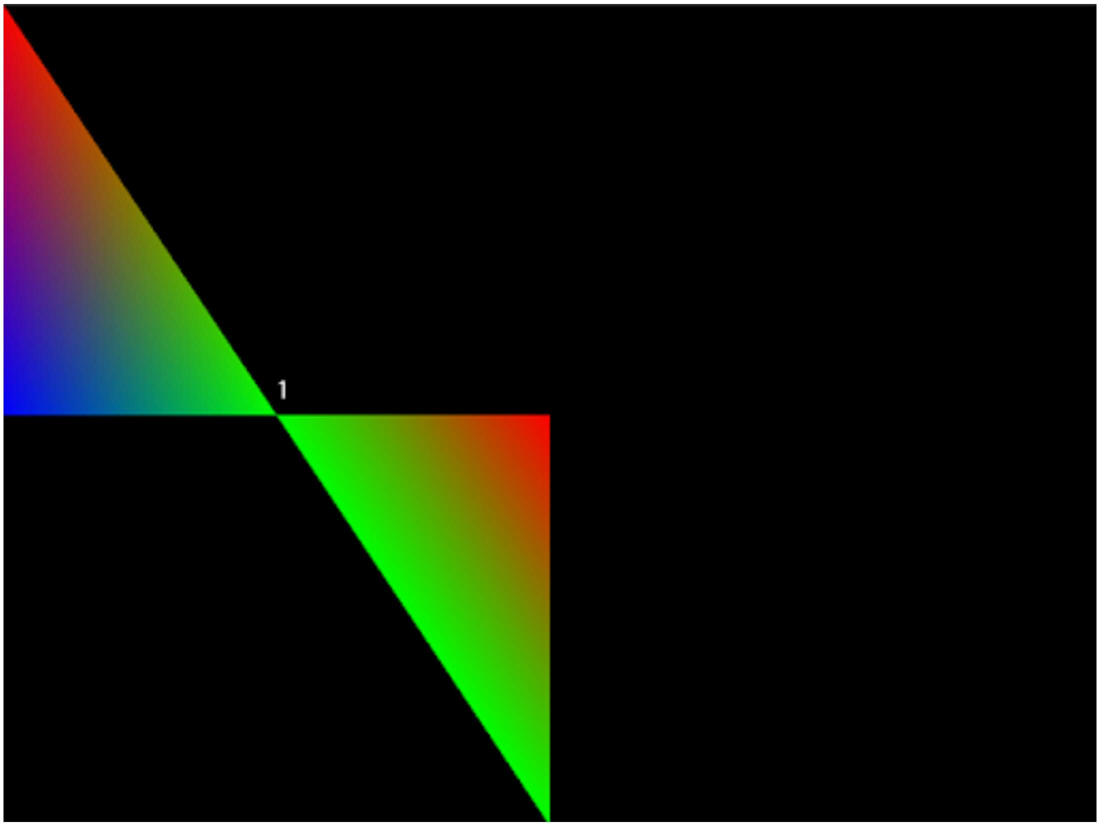
In this assignment, you will create commented code for 2D shapes in modern OpenGL.

If you encounter any challenges while completing this assignment, be sure to post your questions or issues to the General Questions discussion. It is essential to ask for help when you need it and successfully complete each activity, as the course will continue to build on earlier learning as your skills progress.

Prompt

Before you begin, be sure to review the module resources for this week's topics. Then, if you have not done so already, navigate to the CS 330 folder in GitHub. From there, download the ZIP file containing all of the items within this folder, including assignment tutorials for Module Two through Module Six. There are two essential parts to these tutorials: the markdown (MD) files that walk you through how to work with different OpenGL capabilities, and the solution (SLN) file that contains all the code for each section of the tutorials. On your own machine, open the solution (SLN) file in Visual Studio and navigate to the Module Two tutorial sections in the Solution Explorer. While you may open the markdown (MD) file using an external text-based program, we recommend you instead follow along with the Module Two Tutorial in GitHub so it is easier to review the different sections, code, and supporting images. Going through all the sections in the tutorial and attempting the embedded exercises will help you practice the skills you will need to demonstrate in this assignment.

Once you understand the content in the tutorial, you will begin this assignment by opening a new Visual Studio project that has all the libraries set up correctly (which you learned how to do in the previous module). The goal of this assignment is to write commented modern OpenGL code that creates two right-angle triangles.



Specifically, you must address the following rubric criteria:

- **Create code to address the required functionality**. The work you complete in OpenGL must meet the required functionality and visual representation that are outlined for this particular topic. Achieving this result may require multiple attempts or the application of programming strategies, but that is okay! Working in iterations is an important part of any coding project. You may also wish to refer back to relevant sections of this week's tutorial for further guidance or review.

- **Apply logic and proper syntax to code**. Source code should be free of logical or syntax errors that prevent the application from running as expected. You will be given credit for code that is well on its way to meeting specifications or solving the problem.
- **Apply commenting and formatting standards to facilitate understanding of the code**. All code should be well commented. This is a practiced art that requires clarity and concision. Your comments should explain the purpose of lines or sections of the code and may also include the method you used to achieve a specific task in the code. Be sure to document any sections of code that are producing errors or incorrect results. Also, all code should be organized to meet formatting standards.

Guidelines for Submission

Submit a completed ZIP folder with all of your code, which may include one or multiple CPP files along with Visual Studio project files. Also make sure the ZIP folder includes an EXE file, because without this your code will not be able to run. Checking for the EXE can be used as a quick reference on the functionality of your code before you submit. Reference the Visual Studio Export Tutorial, linked in this week's resources, for guidance on how to download the necessary ZIP folder.

Module Two Assignment Rubric

Please note that while this assignment is graded, its main purpose is to provide opportunities for you to build skills and receive feedback that you can incorporate as you work toward the milestones and project.

Criteria	Proficient (100%)	Needs Improvement (75%)	Not Evident (0%)	Value
Required Functionality	Creates code to address required functionality	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include ensuring the details of the functionality or visual assessment are met in significant instances	Does not attempt criterion	50
Code Correctness	Applies logic and proper syntax to code	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include correcting errors in code logic or syntax	Does not attempt criterion	40
Comments and Formatting	Applies commenting and formatting standards to facilitate understanding of the code	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include using complete and clear code comments or consistently following formatting standards	Does not attempt criterion	10
			Total:	100%