2D Geometric Transformations in OpenGL  
Assignment - 2

**Course:** Computer Graphics

**Roll no:** 2023BCS-038

**Name:** Mulpur Aneesh Kumar

**1. About the Assignment**

This assignment is a C program that demonstrates fundamental 2D geometric transformations using the OpenGL graphics library. The application renders a simple 2D object (a triangle) and allows the user to apply various transformations to it in real-time.

The core features of the project are:

* **Interactive Control:** The program is controlled entirely through a console-based menu, where the user can select a transformation and input the required parameters.
* **Smooth Animation:** Instead of instantly applying transformations, the program visualizes the process with smooth animations, making the transition from the initial to the final state clear and easy to understand.
* **Supported Transformations:**
  + Translation (moving the object)
  + Scaling (resizing the object)
  + Rotation (rotating the object about the origin)
  + Reflection (flipping the object across the X or Y axis)
  + Shearing (skewing the object)
* **State Management:** The program allows transformations to be stacked on top of each other and provides a reset option to return the triangle to its original position at any time.

The application is built using C and the OpenGL Utility Toolkit (GLUT) for window management and rendering.

**2. Installation and Execution Guide**

Follow these steps to set up the environment and run the project on a Windows machine using Visual Studio Code.

**Prerequisites**

1. **MinGW-w64 Compiler:** A C/C++ compiler for Windows. It can be downloaded from [winlibs.com](https://winlibs.com/). Ensure its bin directory is added to the system's PATH environment variable.
2. **Visual Studio Code:** With the official Microsoft **C/C++** extension installed.
3. **FreeGLUT Library:** The OpenGL Utility Toolkit is required. The development libraries for MinGW can be downloaded from [Transmission Zero](https://www.transmissionzero.co.uk/software/freeglut-devel/).

**Setup Steps**

1. **Place GLUT Files:** After downloading and extracting FreeGLUT, copy the library files to your MinGW installation directory (e.g., C:\mingw64):
   * Copy the GL folder from freeglut/include to C:\mingw64\include.
   * Copy all files from freeglut/lib/x64 to C:\mingw64\lib.
   * Copy freeglut.dll from freeglut/bin/x64 into the **same folder where the C source code is located**.

**Compilation and Running**

1. **Open Terminal:** Open a terminal (like PowerShell or Command Prompt) in the project directory where the source file (.c) and freeglut.dll are located.
2. **Compile:** Use the following gcc command to compile the program. This links all the necessary graphics libraries.
3. gcc 2d\_transformations.c -o transform.exe -lfreeglut -lopengl32 -lglu32 -lm
4. **Execute:** Run the compiled application from the terminal:
5. ./transform.exe

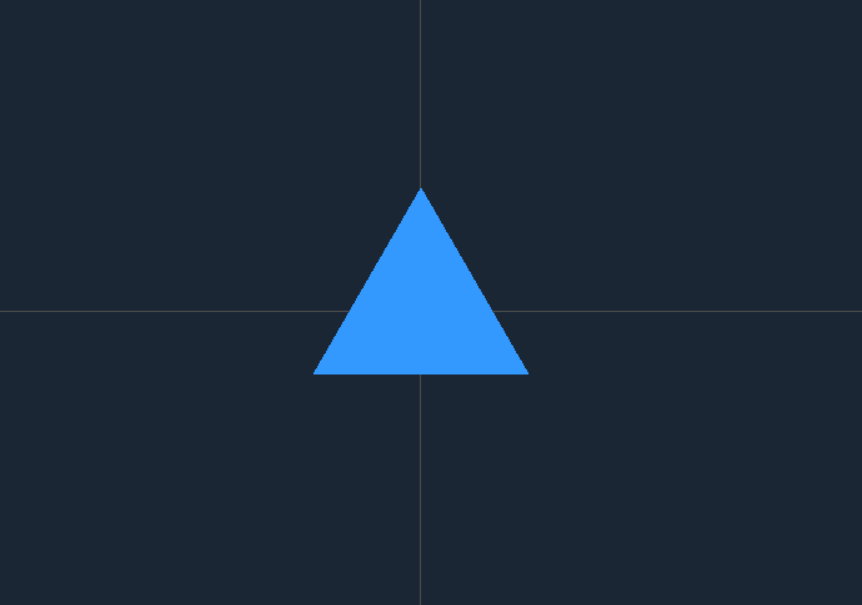
A graphics window will appear, and the control menu will be available in the terminal.

**3. Operations Showcase**

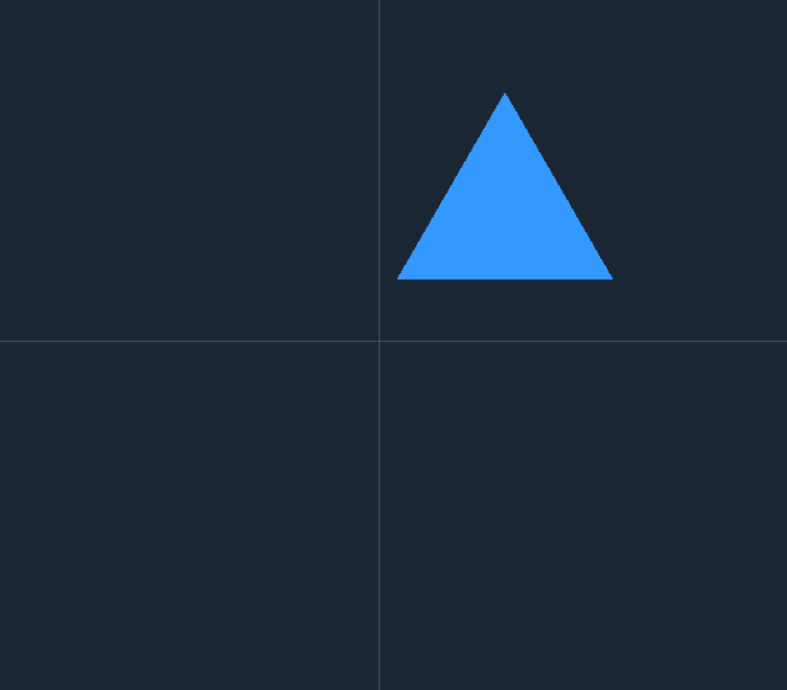
This section demonstrates the output of each transformation applied to the triangle.

**Initial State**

The triangle is initially rendered in the center of the coordinate system.

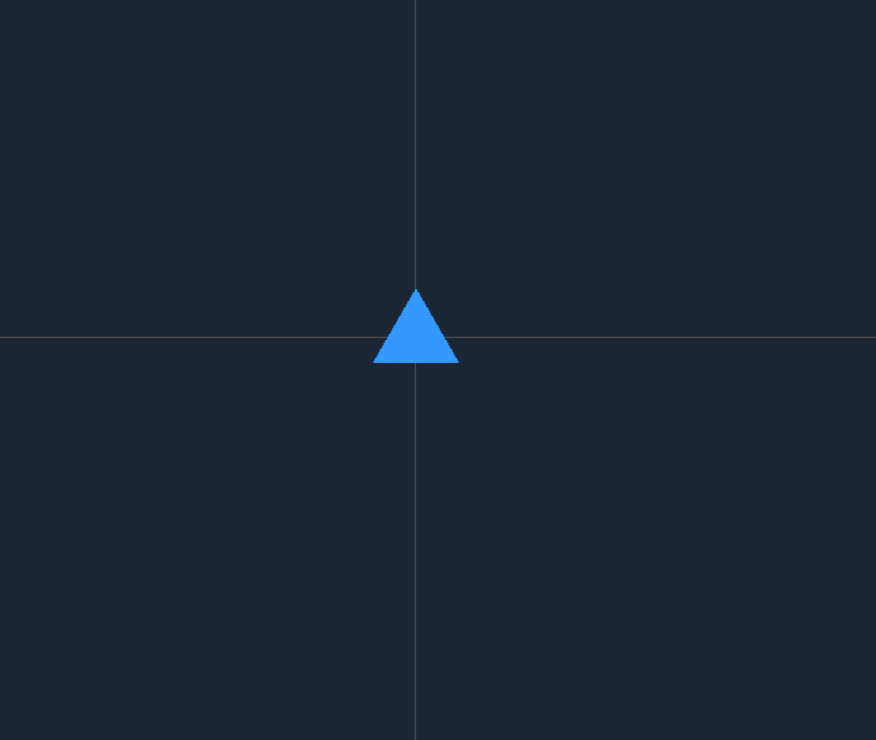
****

**1. Translation**

The triangle is moved to a new position based on the user-provided translation vector (tx, ty).  


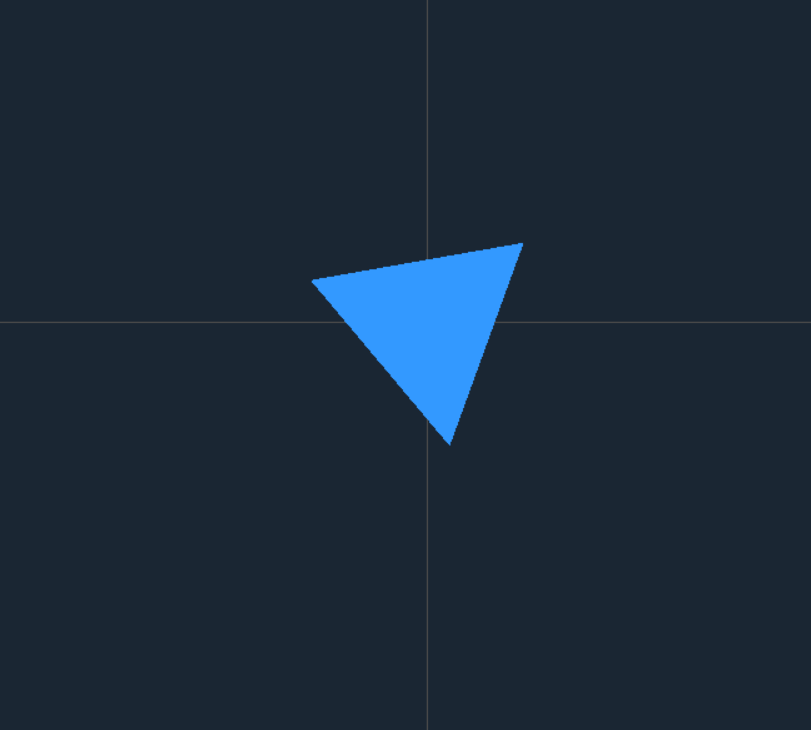
**2. Scaling**

The triangle is resized based on the scaling factors (sx, sy) relative to the origin.

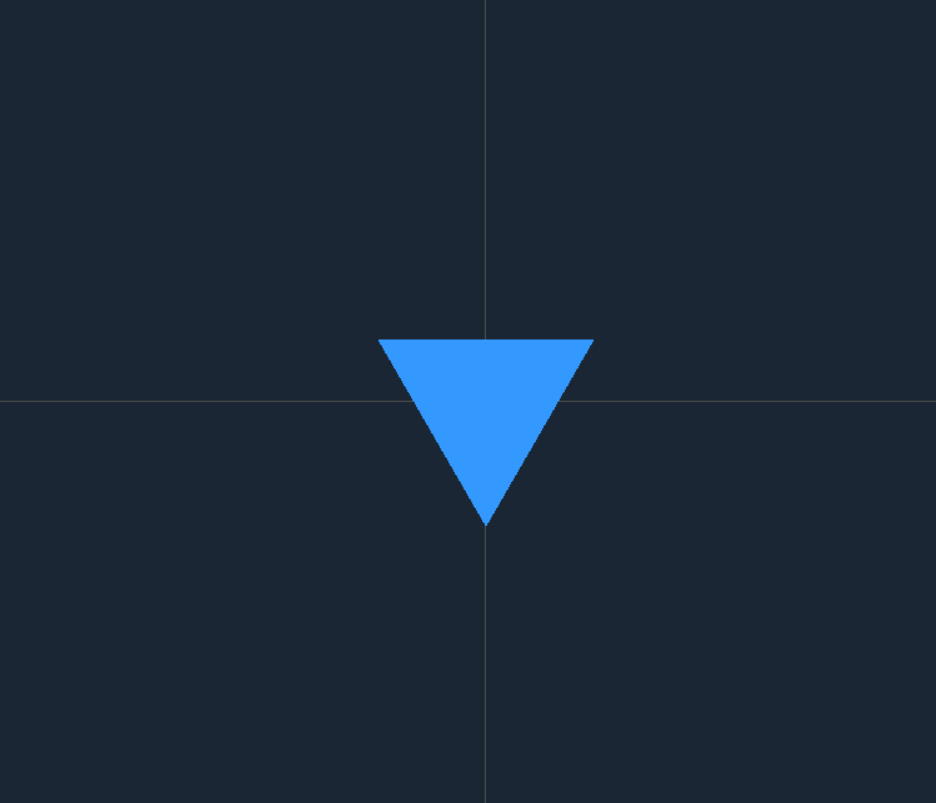


**3. Rotation**

The triangle is rotated around the origin by a specified angle in degrees.



**4. Reflection**

The triangle is reflected based on the axis.  


**5. Shearing**

The triangle is skewed based on the horizontal (shx) and vertical (shy) shear factors.

