Assignment No.6

# Name: Omkar Manohar Hepat

# Class: CS-B

# Roll No: 31 PRN No: 12211509

# Subject: CGAVR

## Q1. Implementation of 3D transformations

#include <GL/glut.h>

GLfloat angle = 0.0f;

GLfloat scale = 1.0f;

GLfloat xTranslation = 0.0f;

GLfloat yTranslation = 0.0f;

GLfloat zTranslation = -5.0f;

GLfloat shearingFactorX = 0.0f;

GLfloat shearingFactorY = 0.0f;

GLboolean reflectX = false;

GLboolean reflectY = false;

void init() {

glClearColor(0.0, 0.0, 0.0, 1.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(45.0, 1.0, 1.0, 100.0);

glMatrixMode(GL\_MODELVIEW);

glEnable(GL\_DEPTH\_TEST);

}

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glLoadIdentity();

glTranslatef(xTranslation, yTranslation, zTranslation);

glRotatef(angle, 1.0f, 1.0f, 1.0f);

glScalef(scale, scale, scale);

if (reflectX)

glScalef(-1.0f, 1.0f, 1.0f);

if (reflectY)

glScalef(1.0f, -1.0f, 1.0f);

glTranslatef(shearingFactorX, shearingFactorY, 0.0f);

glBegin(GL\_QUADS);

glColor3f(1.0f, 0.0f, 0.0f);

glVertex3f(-1.0f, -1.0f, 1.0f);

glVertex3f(1.0f, -1.0f, 1.0f);

glVertex3f(1.0f, 1.0f, 1.0f);

glVertex3f(-1.0f, 1.0f, 1.0f);

glColor3f(0.0f, 1.0f, 0.0f);

glVertex3f(-1.0f, -1.0f, -1.0f);

glVertex3f(-1.0f, 1.0f, -1.0f);

glVertex3f(1.0f, 1.0f, -1.0f);

glVertex3f(1.0f, -1.0f, -1.0f);

glColor3f(0.0f, 0.0f, 1.0f);

glVertex3f(-1.0f, 1.0f, -1.0f);

glVertex3f(-1.0f, 1.0f, 1.0f);

glVertex3f(1.0f, 1.0f, 1.0f);

glVertex3f(1.0f, 1.0f, -1.0f);

glColor3f(1.0f, 1.0f, 0.0f);

glVertex3f(-1.0f, -1.0f, -1.0f);

glVertex3f(1.0f, -1.0f, -1.0f);

glVertex3f(1.0f, -1.0f, 1.0f);

glVertex3f(-1.0f, -1.0f, 1.0f);

glColor3f(1.0f, 0.0f, 1.0f);

glVertex3f(1.0f, -1.0f, -1.0f);

glVertex3f(1.0f, 1.0f, -1.0f);

glVertex3f(1.0f, 1.0f, 1.0f);

glVertex3f(1.0f, -1.0f, 1.0f);

glColor3f(0.0f, 1.0f, 1.0f);

glVertex3f(-1.0f, -1.0f, -1.0f);

glVertex3f(-1.0f, -1.0f, 1.0f);

glVertex3f(-1.0f, 1.0f, 1.0f);

glVertex3f(-1.0f, 1.0f, -1.0f);

glEnd();

glFlush();

glutSwapBuffers();

}

void reshape(int w, int h) {

glViewport(0, 0, (GLsizei)w, (GLsizei)h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(45.0, (GLfloat)w / (GLfloat)h, 1.0, 100.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

void specialKeys(int key, int x, int y) {

switch (key) {

case GLUT\_KEY\_RIGHT:

angle += 5;

break;

case GLUT\_KEY\_LEFT:

angle -= 5;

break;

case GLUT\_KEY\_UP:

scale += 0.1;

break;

case GLUT\_KEY\_DOWN:

scale -= 0.1;

break;

case GLUT\_KEY\_PAGE\_UP:

yTranslation += 0.1;

break;

case GLUT\_KEY\_PAGE\_DOWN:

yTranslation -= 0.1;

break;

case GLUT\_KEY\_HOME:

xTranslation -= 0.1;

break;

case GLUT\_KEY\_END:

xTranslation += 0.1;

break;

case 127:

zTranslation += 0.1;

break;

}

glutPostRedisplay();

}

void keyboard(unsigned char key, int x, int y) {

switch (key) {

case 'x':

reflectX = !reflectX;

break;

case 'y':

reflectY = !reflectY;

break;

case 's':

shearingFactorX += 0.1;

break;

case 'S':

shearingFactorX -= 0.1;

break;

case 't':

shearingFactorY += 0.1;

break;

case 'T':

shearingFactorY -= 0.1;

break;

}

glutPostRedisplay();

}

int main(int argc, char\*\* argv) {

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB | GLUT\_DEPTH);

glutInitWindowSize(500, 500);

glutInitWindowPosition(100, 100);

glutCreateWindow("3D Transformations");

init();

glutDisplayFunc(display);

glutReshapeFunc(reshape);

glutSpecialFunc(specialKeys);

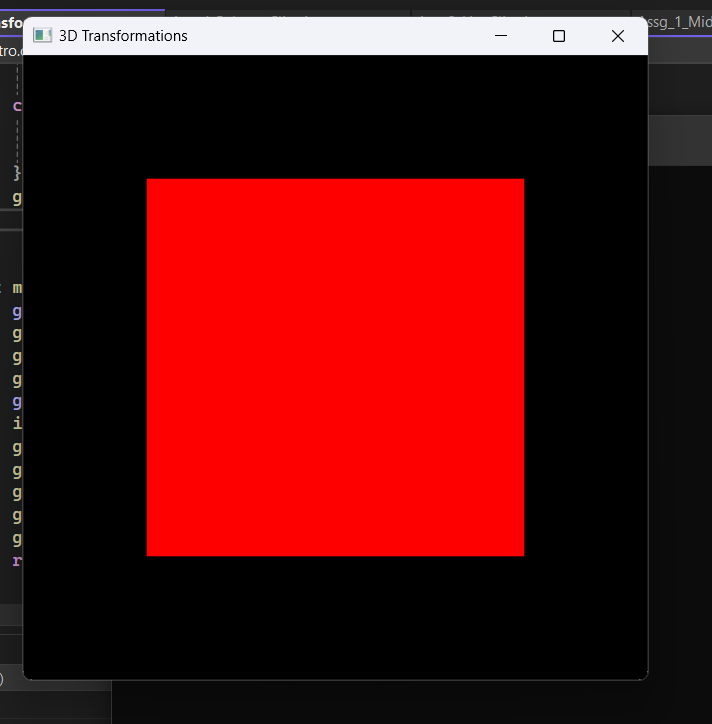
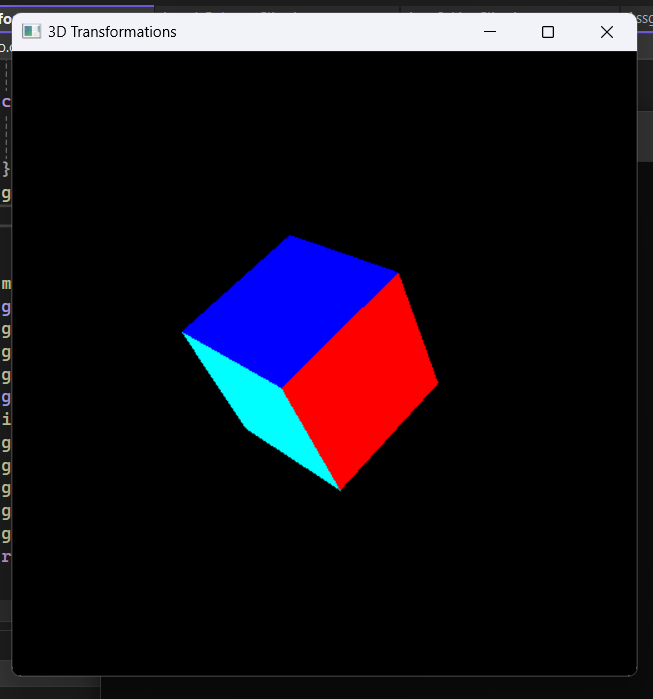
glutKeyboardFunc(keyboard);

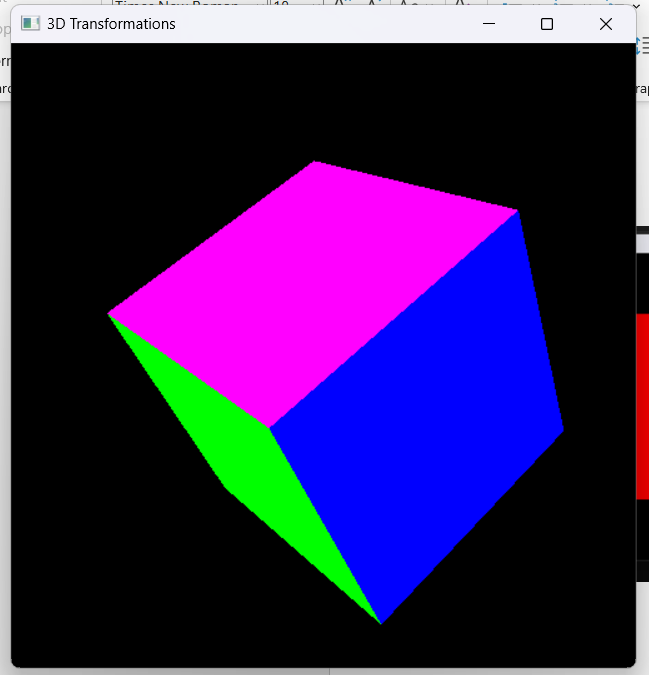
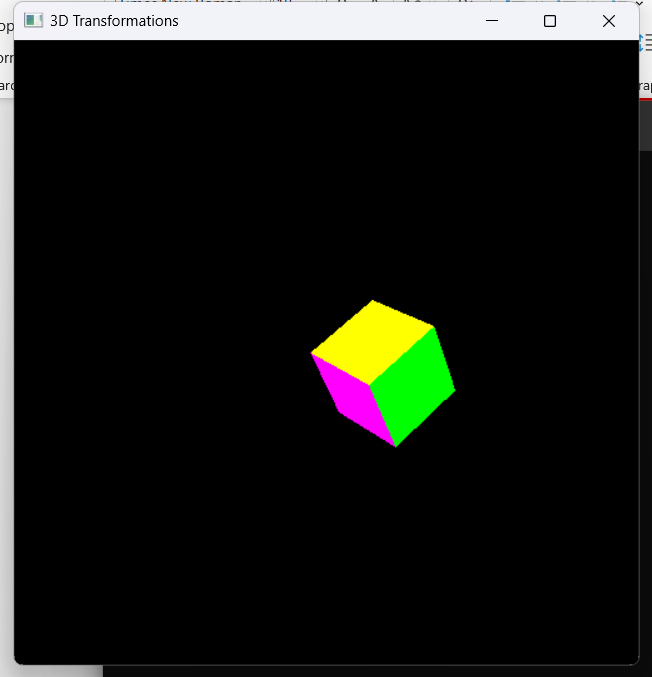
glutMainLoop();

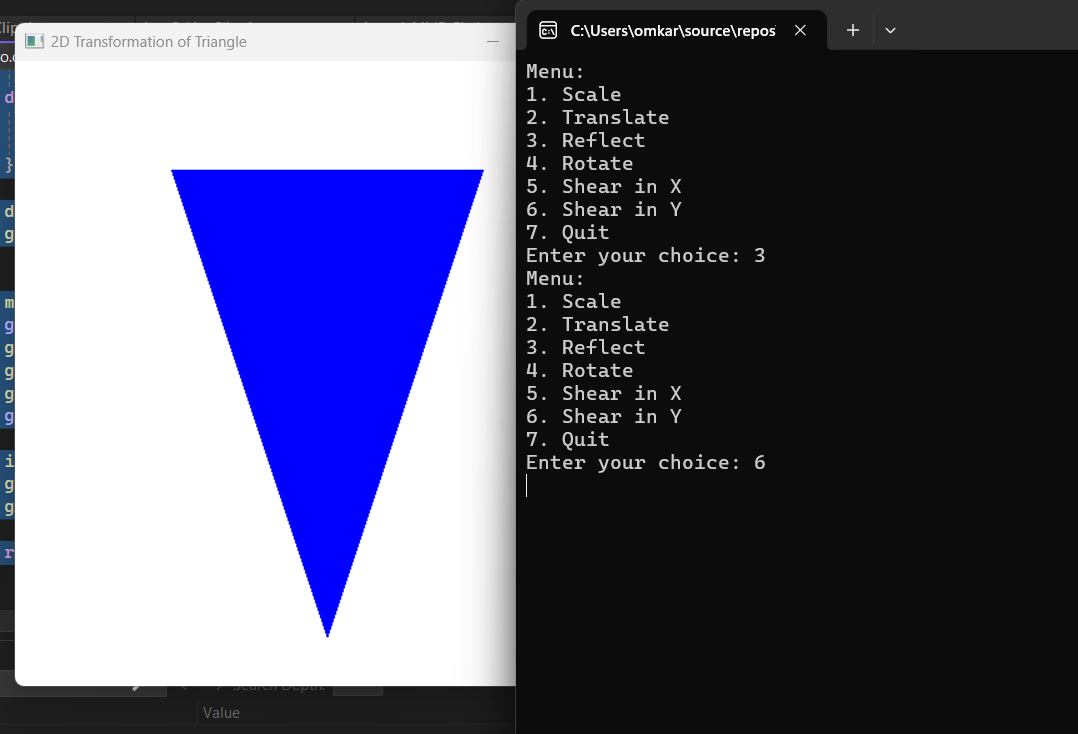
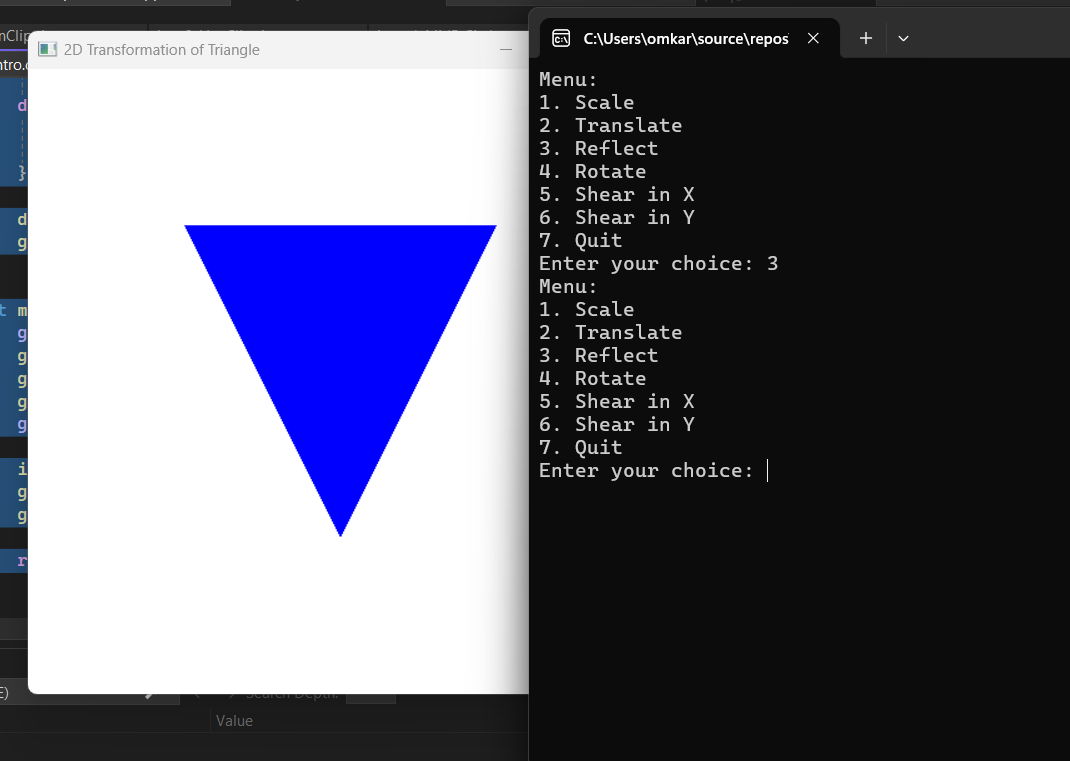
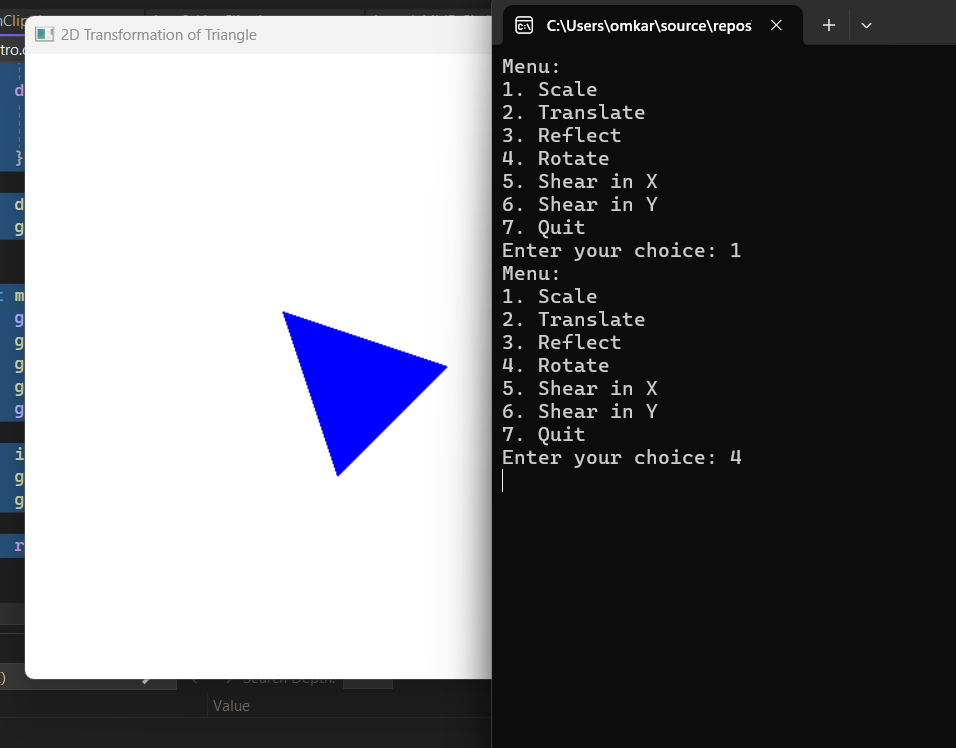
return 0;

}

Output:

****