6. Implement following 2D transformations on the object with respect to axis

i) Scaling ii) Rotation about arbitrary point iii) Reflection.

Source Code :

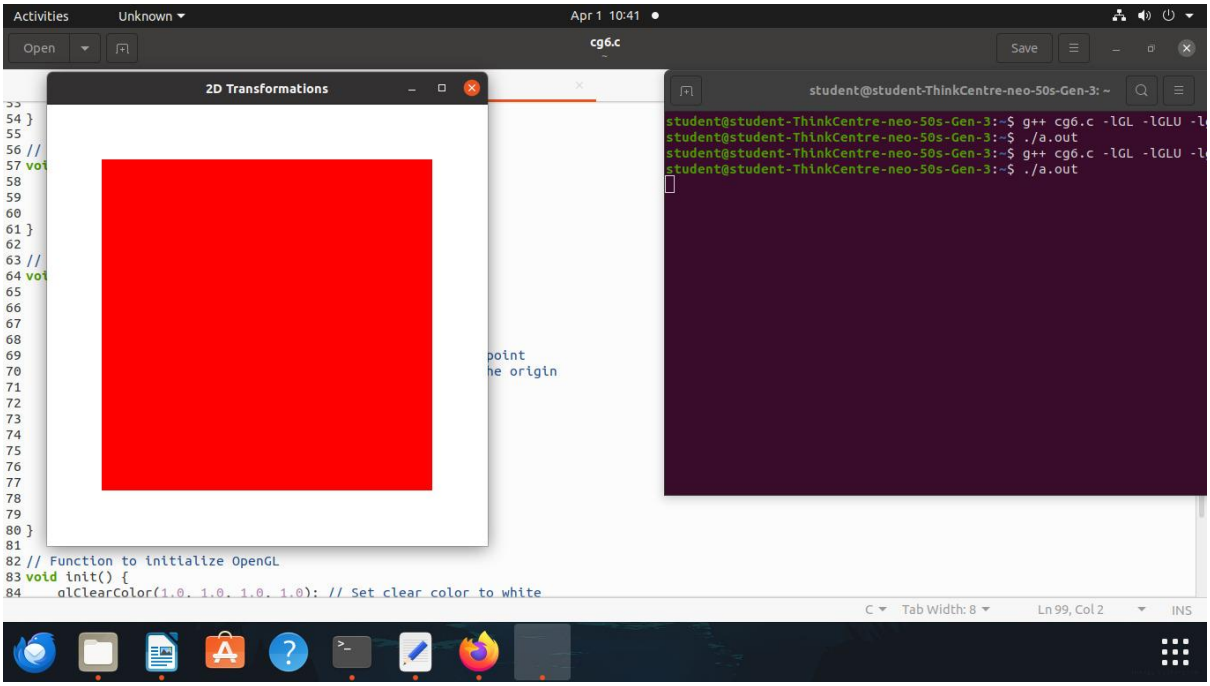
```
#include <GL/glut.h>
#include <stdio.h>
#include <math.h>

float vertices[4][2] = {
    {-25, -25}, // Bottom-left
    {25, -25},  // Bottom-right
    {25, 25},   // Top-right
    {-25, 25}   // Top-left
};

void drawObject() {
    glBegin(GL_POLYGON);
    glColor3f(1.0, 0.0, 0.0); // Set color to red
    for (int i = 0; i < 4; i++) {
        glVertex2fv(vertices[i]);
    }
    glEnd();
}

void scale(float sx, float sy) {
    for (int i = 0; i < 4; i++) {
        vertices[i][0] *= sx;
```

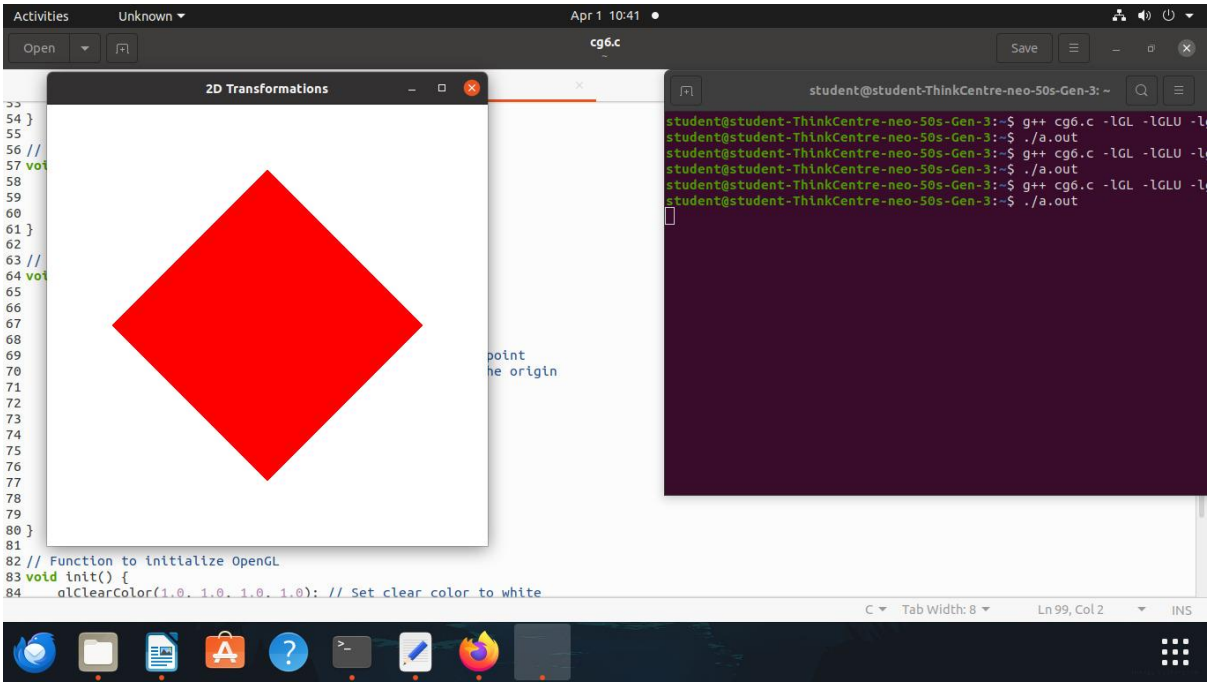
```
        vertices[i][1] *= sy;
    }
}

void rotate(float angle, float px, float py) {
    float radians = angle * M_PI / 180.0;
    for (int i = 0; i < 4; i++) {
        float x = vertices[i][0];
        float y = vertices[i][1];
        vertices[i][0] = px + (x - px) * cos(radians) - (y - py) * sin(radians);
        vertices[i][1] = py + (x - px) * sin(radians) + (y - py) * cos(radians);
    }
}

void reflect(int axis) {
    // Axis: 0 - X-axis, 1 - Y-axis
    if (axis == 0) { // Reflection about X-axis
        for (int i = 0; i < 4; i++) {
            vertices[i][1] *= -1;
        }
    } else if (axis == 1) { // Reflection about Y-axis
        for (int i = 0; i < 4; i++) {
            vertices[i][0] *= -1;
        }
    }
}

void display() {
    glClear(GL_COLOR_BUFFER_BIT);
    drawObject();
```

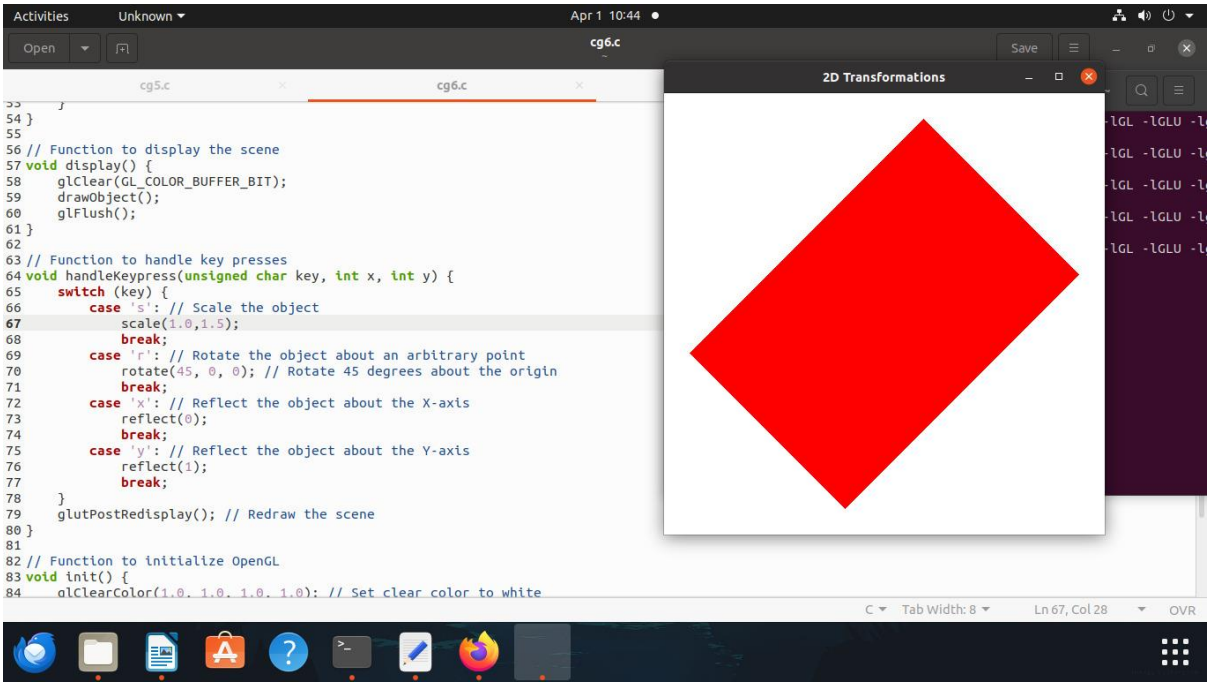
```
    glFlush();
}

void handleKeypress(unsigned char key, int x, int y) {
    switch (key) {
        case 's': // Scale the object
            scale(1.5, 1.5);
            break;
        case 'r': // Rotate the object about an arbitrary point
            rotate(45, 0, 0); // Rotate 45 degrees about the origin
            break;
        case 'x': // Reflect the object about the X-axis
            reflect(0);
            break;
        case 'y': // Reflect the object about the Y-axis
            reflect(1);
            break;
    }
    glutPostRedisplay(); // Redraw the scene
}

void init() {
    glClearColor(1.0, 1.0, 1.0, 1.0); // Set clear color to white
    gluOrtho2D(-50, 50, -50, 50); // Set orthographic projection
}

int main(int argc, char **argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500, 500); // Set window size
```

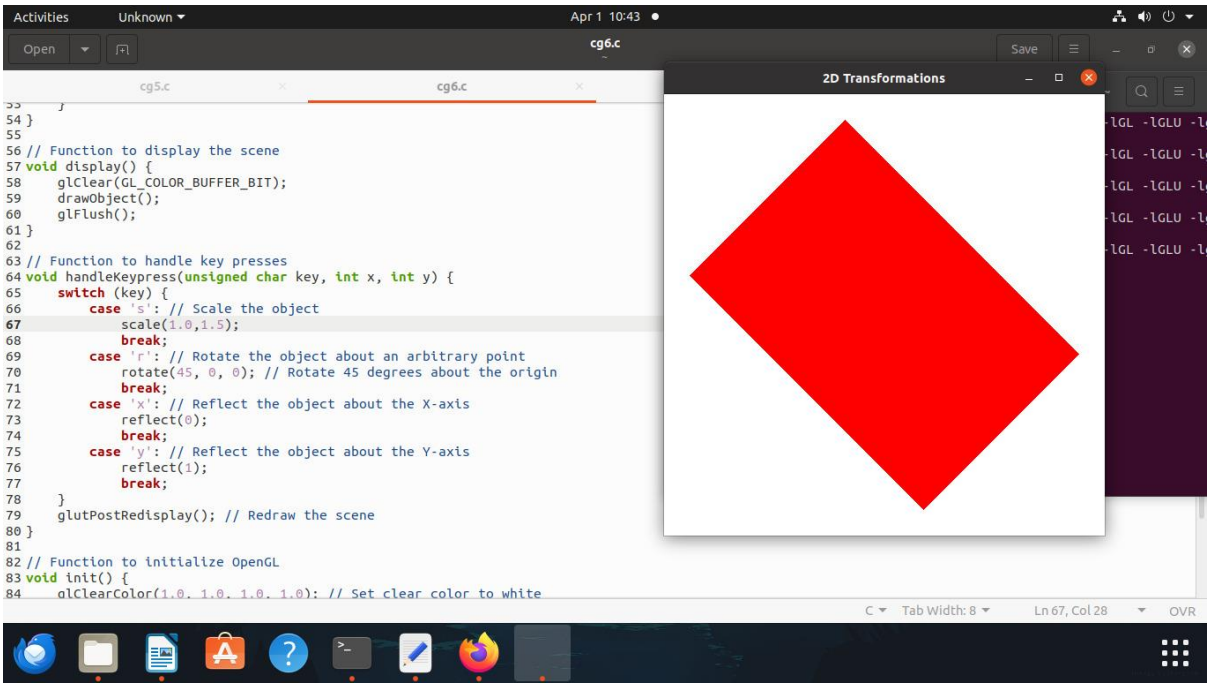
```
    glutCreateWindow("2D Transformations");
    init();
    glutDisplayFunc(display);
    glutKeyboardFunc(handleKeypress);
    glutMainLoop();
    return 0;
}
```

Output :


Activities
Unknown
Apr 1 10:41
Open
cg6.c
Save
2D Transformations
student@student-ThinkCentre-neo-50s-Gen-3: ~
student@student-ThinkCentre-neo-50s-Gen-3:~$ g++ cg6.c -lGL -lGLU -l
student@student-ThinkCentre-neo-50s-Gen-3:~$ ./a.out
student@student-ThinkCentre-neo-50s-Gen-3:~$ g++ cg6.c -lGL -lGLU -l
student@student-ThinkCentre-neo-50s-Gen-3:~$ ./a.out
// Function to initialize OpenGL
void init() {
glClearColor(1.0, 1.0, 1.0, 1.0); // Set clear color to white
C
Tab Width: 8
Ln 99, Col 2
INS


Activities
Unknown
Apr 1 10:41
Open
cg6.c
Save
2D Transformations
student@student-ThinkCentre-neo-50s-Gen-3: ~
student@student-ThinkCentre-neo-50s-Gen-3:~$ g++ cg6.c -lGL -lGLU -l
student@student-ThinkCentre-neo-50s-Gen-3:~$ ./a.out
student@student-ThinkCentre-neo-50s-Gen-3:~$ g++ cg6.c -lGL -lGLU -l
student@student-ThinkCentre-neo-50s-Gen-3:~$ ./a.out
student@student-ThinkCentre-neo-50s-Gen-3:~$ g++ cg6.c -lGL -lGLU -l
student@student-ThinkCentre-neo-50s-Gen-3:~$ ./a.out
// Function to initialize OpenGL
void init() {
glClearColor(1.0, 1.0, 1.0, 1.0); // Set clear color to white
C
Tab Width: 8
Ln 99, Col 2
INS


Activities
Unknown
Apr 1 10:44
Open
cg6.c
Save
cg5.c
cg6.c
2D Transformations
54 }
55
56 // Function to display the scene
57 void display() {
58 glClear(GL_COLOR_BUFFER_BIT);
59 drawObject();
60 glFlush();
61 }
62
63 // Function to handle key presses
64 void handleKeypress(unsigned char key, int x, int y) {
65 switch (key) {
66 case 's': // Scale the object
67 scale(1.0,1.5);
68 break;
69 case 'r': // Rotate the object about an arbitrary point
70 rotate(45, 0, 0); // Rotate 45 degrees about the origin
71 break;
72 case 'x': // Reflect the object about the X-axis
73 reflect(0);
74 break;
75 case 'y': // Reflect the object about the Y-axis
76 reflect(1);
77 break;
78 }
79 glutPostRedisplay(); // Redraw the scene
80 }
81
82 // Function to initialize OpenGL
83 void init() {
84 glClearColor(1.0, 1.0, 1.0, 1.0); // Set clear color to white
C
Tab Width: 8
Ln 67, Col 28
OVR


Activities
Unknown
Apr 1 10:43
Open
cg6.c
Save
cg5.c
cg6.c
2D Transformations
54 }
55
56 // Function to display the scene
57 void display() {
58 glClear(GL_COLOR_BUFFER_BIT);
59 drawObject();
60 glFlush();
61 }
62
63 // Function to handle key presses
64 void handleKeypress(unsigned char key, int x, int y) {
65 switch (key) {
66 case 's': // Scale the object
67 scale(1.0,1.5);
68 break;
69 case 'r': // Rotate the object about an arbitrary point
70 rotate(45, 0, 0); // Rotate 45 degrees about the origin
71 break;
72 case 'x': // Reflect the object about the X-axis
73 reflect(0);
74 break;
75 case 'y': // Reflect the object about the Y-axis
76 reflect(1);
77 break;
78 }
79 glutPostRedisplay(); // Redraw the scene
80 }
81
82 // Function to initialize OpenGL
83 void init() {
84 glClearColor(1.0, 1.0, 1.0, 1.0); // Set clear color to white
C
Tab Width: 8
Ln 67, Col 28
OVR