**LAB-12**

OPENGL app to to rotate the rectangle using opengl function

#include <GL/glut.h>

#include <iostream>

using namespace std;

int rectWidth=200,rectHeight=50;

int oneX=100,oneY=100;

int vertices[ ][4]={{oneX,oneX+rectWidth,oneX+rectWidth,oneX},

{oneY,oneY,oneY+rectHeight,oneY+rectHeight},

{1,1,1,1}};

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(1, 1, 1, 1);

glColor3d(0,0,0);

glBegin(GL\_LINE\_LOOP);

for(int i=0;i<4;i++){

glVertex2i(vertices[0][i],vertices[1][i]);

}

glEnd();

glRotated(30,0,0,1);

glBegin(GL\_LINE\_LOOP);

for(int i=0;i<4;i++){

glVertex2i(vertices[0][i],vertices[1][i]);

}

glEnd();

glFlush();

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_RGB);

glutInitWindowSize(1024,768);

glutCreateWindow("LAB 12 Niraj Maharjan");

glClearColor(1, 1, 1, 1);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-500,500,-500,500);

glMatrixMode(GL\_MODELVIEW);

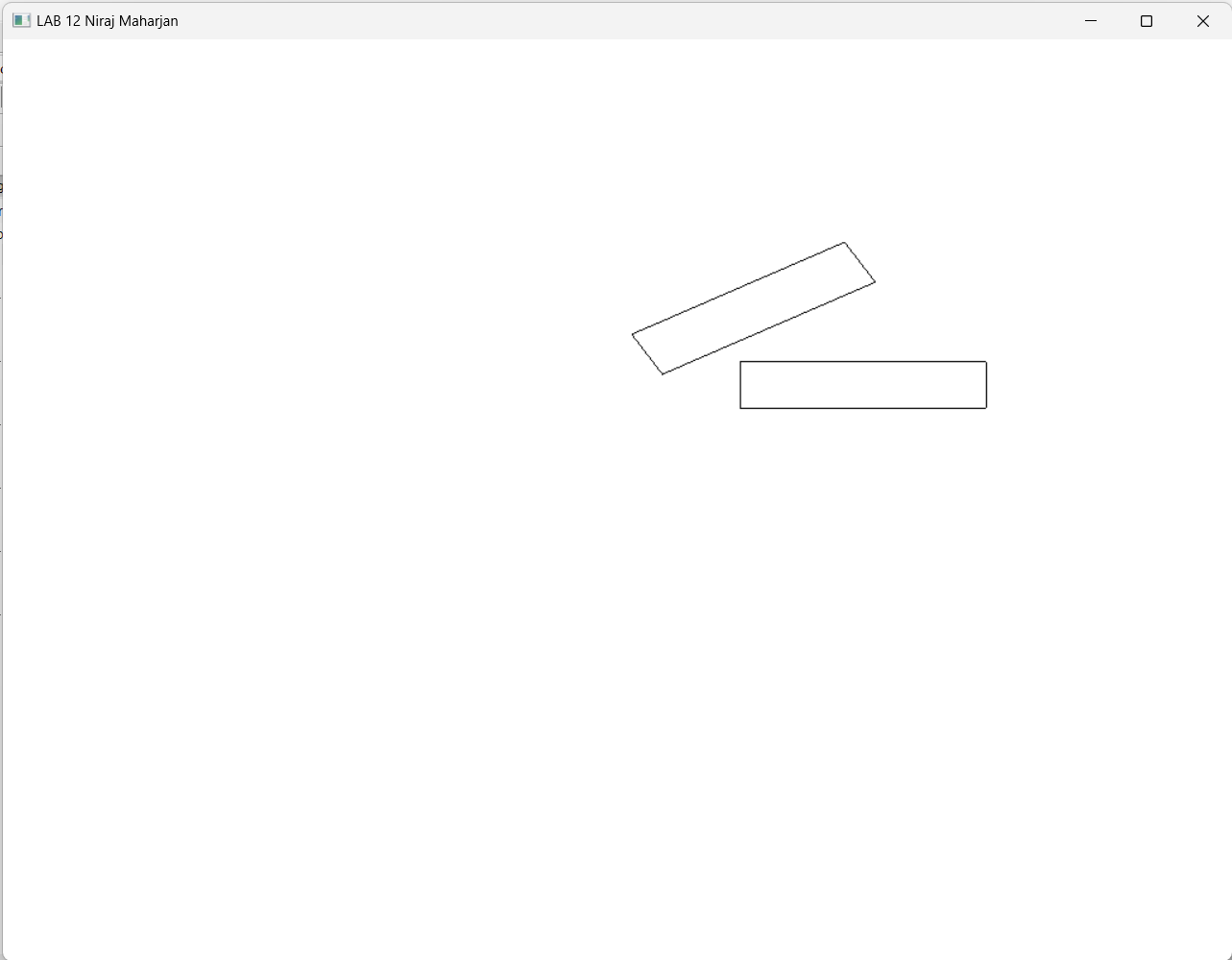
glutDisplayFunc(display);

glutMainLoop();

return 0;

}

**OUTPUT**



**LAB-13**

OPENGL app to to rotate the rectangle without using opengl function

#include <GL/glut.h>

#include <iostream>

#include <cmath>

using namespace std;

int rectWidth=200,rectHeight=50;

int oneX=100,oneY=100;

int vertices[][4]={{oneX,oneX+rectWidth,oneX+rectWidth,oneX},

{oneY,oneY,oneY+rectHeight,oneY+rectHeight},

{1,1,1,1}};

void rotateVertices(int angle) {

double angleRad = angle \* M\_PI / 180.0;

for (int i = 0; i < 4; i++) {

int x = vertices[0][i];

int y = vertices[1][i];

vertices[0][i] = round(x \* cos(angleRad) - y \* sin(angleRad));

vertices[1][i] = round(x \* sin(angleRad) + y \* cos(angleRad));

}

}

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(1, 1, 1, 1);

glColor3d(0,0,0);

glBegin(GL\_LINE\_LOOP);

for(int i=0;i<4;i++){

glVertex2i(vertices[0][i],vertices[1][i]);

}

glEnd();

rotateVertices(30);

glBegin(GL\_LINE\_LOOP);

for(int i=0;i<4;i++){

glVertex2i(vertices[0][i],vertices[1][i]);

}

glEnd();

glFlush();

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_RGB);

glutInitWindowSize(1024,768);

glutCreateWindow("LAB 13 Niraj Maharjan");

glClearColor(1, 1, 1, 1);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-500,500,-500,500);

glMatrixMode(GL\_MODELVIEW);

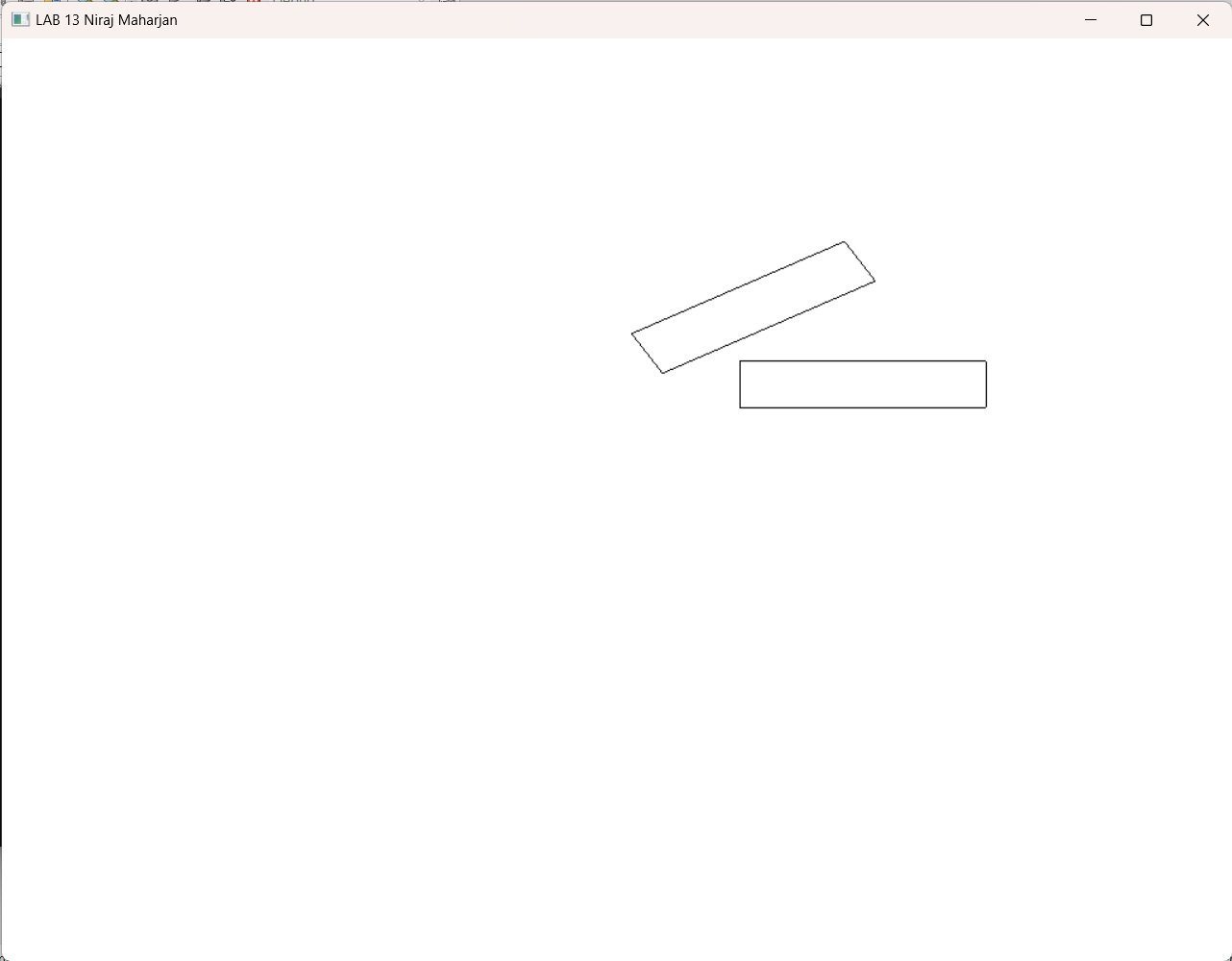
glutDisplayFunc(display);

glutMainLoop();

return 0;

}

**OUTPUT**

****

**LAB-14**

OPENGL app to scale the rectangle by (2,2) using opengl function

#include <GL/glut.h>

#include <iostream>

using namespace std;

int rectWidth=200,rectHeight=50;

int oneX=100,oneY=100;

int vertices[][4]={{oneX,oneX+rectWidth,oneX+rectWidth,oneX},

{oneY,oneY,oneY+rectHeight,oneY+rectHeight},

{1,1,1,1}};

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(1, 1, 1, 1);

glColor3d(0,0,0);

glBegin(GL\_LINE\_LOOP);

for(int i=0;i<4;i++){

glVertex2i(vertices[0][i],vertices[1][i]);

}

glEnd();

glScaled(2,2,0);

glBegin(GL\_LINE\_LOOP);

for(int i=0;i<4;i++){

glVertex2i(vertices[0][i],vertices[1][i]);

}

glEnd();

glFlush();

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_RGB);

glutInitWindowSize(1024,768);

glutCreateWindow("LAB 14 Niraj Maharjan");

glClearColor(1, 1, 1, 1);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-1000,1000,-1000,1000);

glMatrixMode(GL\_MODELVIEW);

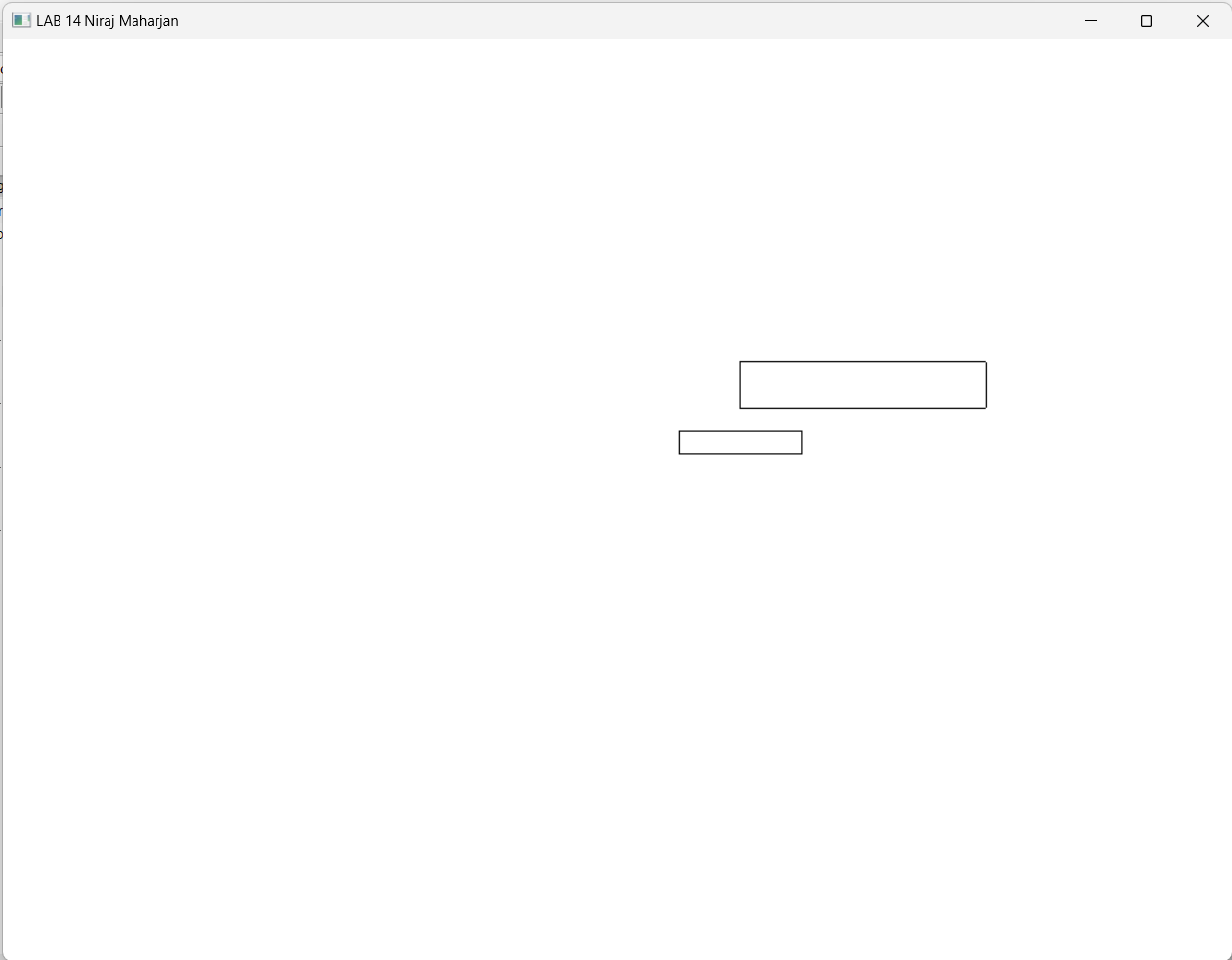
glutDisplayFunc(display);

glutMainLoop();

return 0;

}

**OUTPUT**

****

**LAB-15**

OPENGL app to scale the rectangle by (2,2) without using opengl function

#include <GL/glut.h>

#include <iostream>

using namespace std;

int rectWidth=200,rectHeight=50;

int oneX=100,oneY=100;

int vertices[ ][4]={{oneX,oneX+rectWidth,oneX+rectWidth,oneX},

{oneY,oneY,oneY+rectHeight,oneY+rectHeight},

{1,1,1,1}};

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(1, 1, 1, 1);

glColor3d(0,0,0);

glBegin(GL\_LINE\_LOOP);

for(int i=0;i<4;i++){

glVertex2i(vertices[0][i],vertices[1][i]);

}

glEnd();

glBegin(GL\_LINE\_LOOP);

for (int i = 0; i < 4; i++) {

int scaledX = vertices[0][i] \* 2;

int scaledY = vertices[1][i] \* 2;

glVertex2i(scaledX, scaledY);

}

glEnd();

glFlush();

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_RGB);

glutInitWindowSize(1024,768);

glutCreateWindow("LAB 15 Niraj Maharjan");

glClearColor(1, 1, 1, 1);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-1000,1000,-1000,1000);

glMatrixMode(GL\_MODELVIEW);

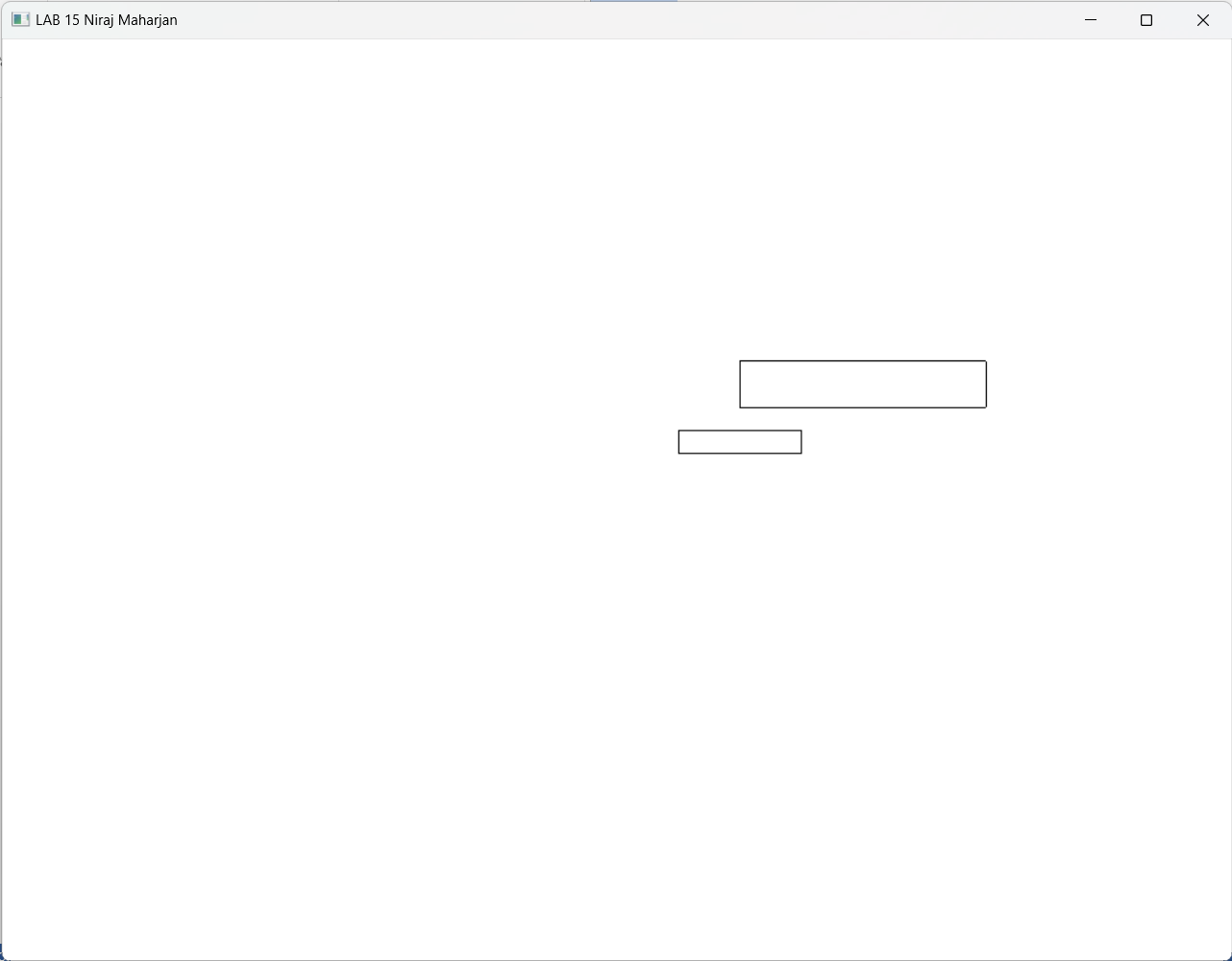
glutDisplayFunc(display);

glutMainLoop();

return 0;

}

**OUTPUT**

****

**LAB-16**

OPENGL app to rotate the rectangle about (150,125) using opengl function

#include <GL/glut.h>

#include <iostream>

using namespace std;

int rectWidth=200,rectHeight=50;

int oneX=100,oneY=100;

int vertices[][4]={{oneX,oneX+rectWidth,oneX+rectWidth,oneX},

{oneY,oneY,oneY+rectHeight,oneY+rectHeight},

{1,1,1,1}};

int centerX=150,centerY=125;

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(1, 1, 1, 1);

glColor3d(0,0,0);

glBegin(GL\_LINE\_LOOP);

for(int i=0;i<4;i++){

glVertex2i(vertices[0][i],vertices[1][i]);

}

glEnd();

glTranslated(-centerX,-centerY,0);

glRotated(30,0,0,1);

glTranslated(centerX,centerY,0);

glBegin(GL\_LINE\_LOOP);

for(int i=0;i<4;i++){

glVertex2i(vertices[0][i],vertices[1][i]);

}

glEnd();

glFlush();

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_RGB);

glutInitWindowSize(1024,768);

glutCreateWindow("LAB 16 Niraj Maharjan");

glClearColor(1, 1, 1, 1);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-500,500,-500,500);

glMatrixMode(GL\_MODELVIEW);

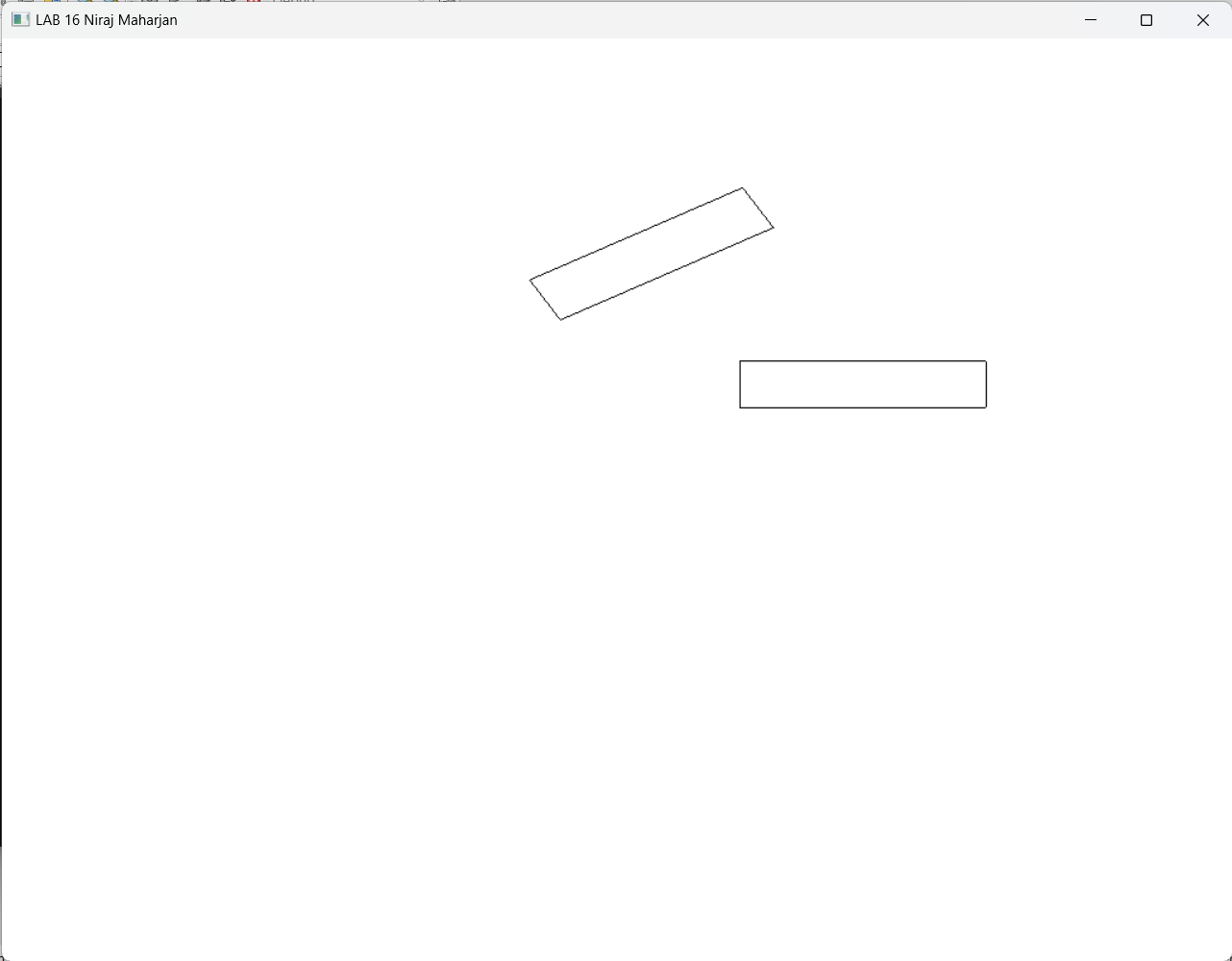
glutDisplayFunc(display);

glutMainLoop();

return 0;

}

**OUTPUT**



**LAB-17**

OPENGL app to rotate the rectangle about (150,125) without using opengl function

#include <GL/glut.h>

#include <iostream>

#include <cmath>

using namespace std;

int rectWidth=200,rectHeight=50;

int oneX=100,oneY=100;

int vertices[][4]={{oneX,oneX+rectWidth,oneX+rectWidth,oneX},

{oneY,oneY,oneY+rectHeight,oneY+rectHeight},

{1,1,1,1}};

int centerX=150,centerY=125;

void rotateVertices(int angle) {

double angleRad = angle \* M\_PI / 180.0;

for (int i = 0; i < 4; i++) {

int x = vertices[0][i];

int y = vertices[1][i];

vertices[0][i] = round(x \* cos(angleRad) - y \* sin(angleRad));

vertices[1][i] = round(x \* sin(angleRad) + y \* cos(angleRad));

}

}

void translateVertices(int tx,int ty){

for (int i = 0; i < 4; i++) {

int x = vertices[0][i];

int y = vertices[1][i];

vertices[0][i] = x+tx;

vertices[1][i] = y+ty;

}

}

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(1, 1, 1, 1);

glColor3d(0,0,0);

glBegin(GL\_LINE\_LOOP);

for(int i=0;i<4;i++){

glVertex2i(vertices[0][i],vertices[1][i]);

}

glEnd();

translateVertices(-centerX,-centerY);

rotateVertices(30);

translateVertices(centerX,centerY);

glBegin(GL\_LINE\_LOOP);

for(int i=0;i<4;i++){

glVertex2i(vertices[0][i],vertices[1][i]);

}

glEnd();

glFlush();

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_RGB);

glutInitWindowSize(1024,768);

glutCreateWindow("LAB 17 Niraj Maharjan");

glClearColor(1, 1, 1, 1);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

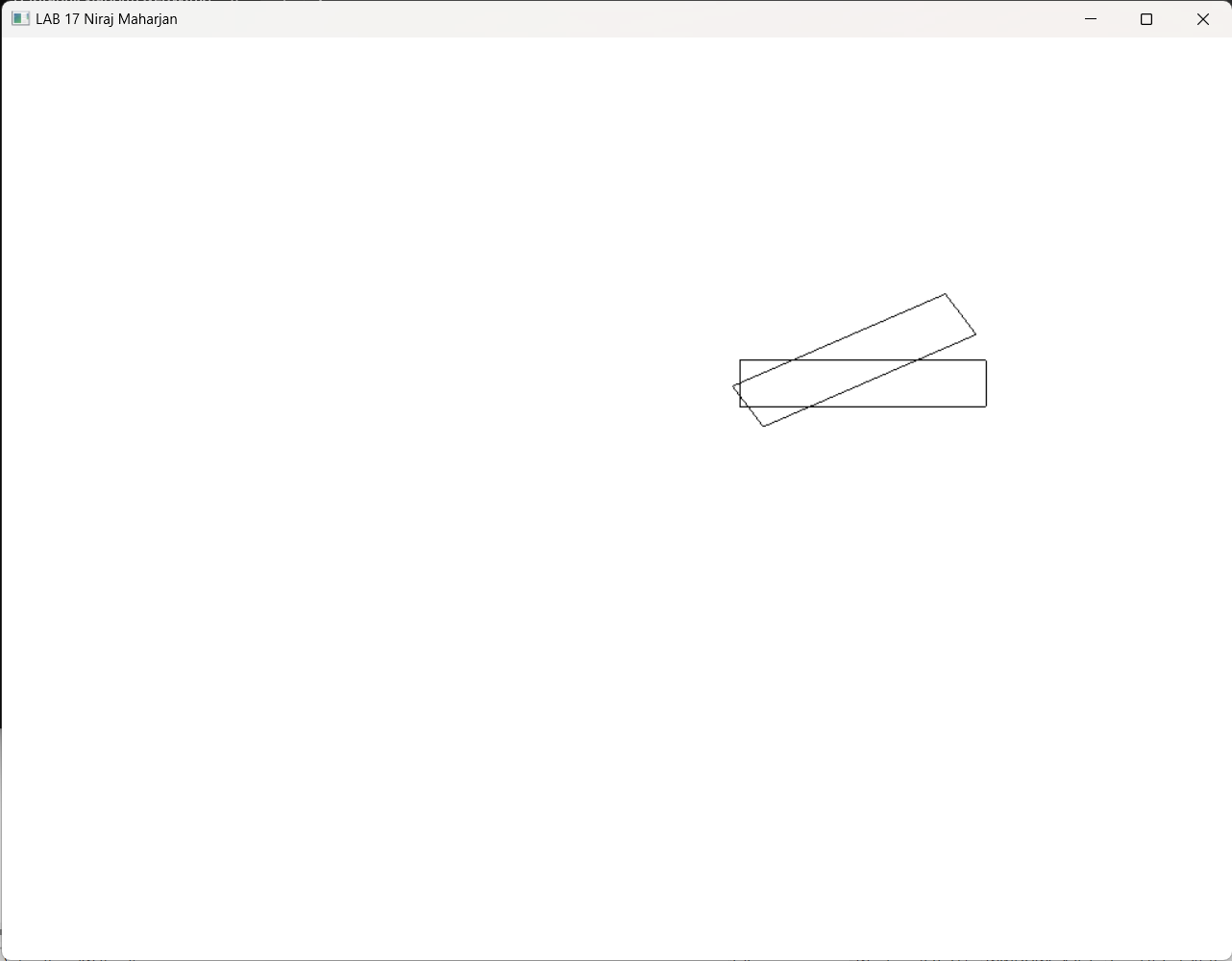
gluOrtho2D(-500,500,-500,500);

glMatrixMode(GL\_MODELVIEW);

glutDisplayFunc(display);

glutMainLoop();

return 0;

}

**OUTPUT**