**SSN COLLEGE OF ENGINEERING, KALAVAKKAM**

}

**void** myDisplay() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_POINTS);

glVertex2d(150,100);

glVertex2d(230,100);

glVertex2d(120,170);

glVertex2d(300,350);

glEnd();

glFlush();

}

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

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(640,480);

glutCreateWindow("First Exercise");

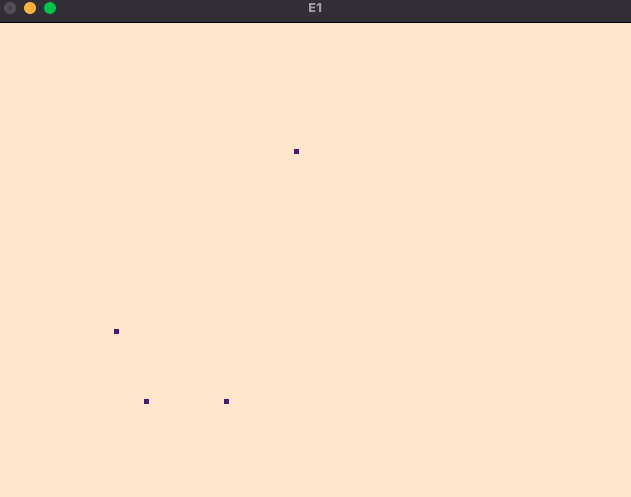
glutDisplayFunc(myDisplay);

myInit();

glutMainLoop();

**return** 1;

}



LINES:

#include<GLUT/glut.h>

**void** myInit() {

glClearColor(1.0,0.9,0.8,0.5);

glColor3f(0.1f,0.1f,0.1f);

glPointSize(5);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0,640.0,0.0,480.0);

}

**void** myDisplay() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_LINES);

glVertex2d(150,100);

glVertex2d(170,200);

glVertex2d(200,130);

glVertex2d(310,350);

glEnd();

glFlush();

}

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

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(640,480);

glutCreateWindow("First Exercise");

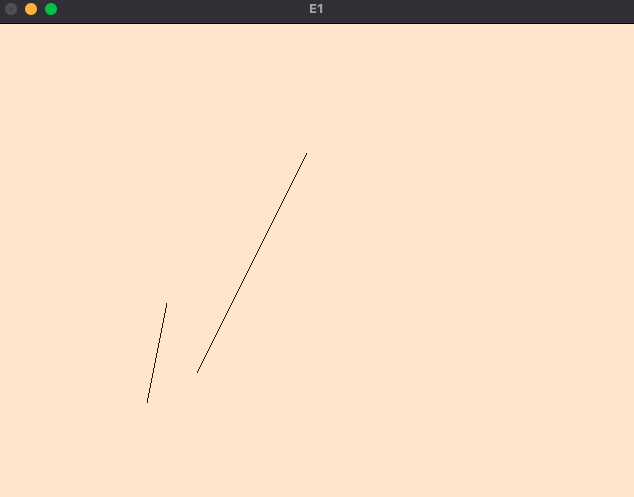
glutDisplayFunc(myDisplay);

myInit();

glutMainLoop();

**return** 1;

}

**

LINE\_STRIP:

#include<GLUT/glut.h>

**void** myInit() {

glClearColor(1.0,0.9,0.8,0.5);

glColor3f(0.1f,0.1f,0.1f);

glPointSize(5);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0,640.0,0.0,480.0);

}

**void** myDisplay() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_LINE\_STRIP);

glVertex2d(150,100);

glVertex2d(170,200);

glVertex2d(200,130);

glVertex2d(310,350);

glEnd();

glFlush();

}

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

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(640,480);

glutCreateWindow("E1");

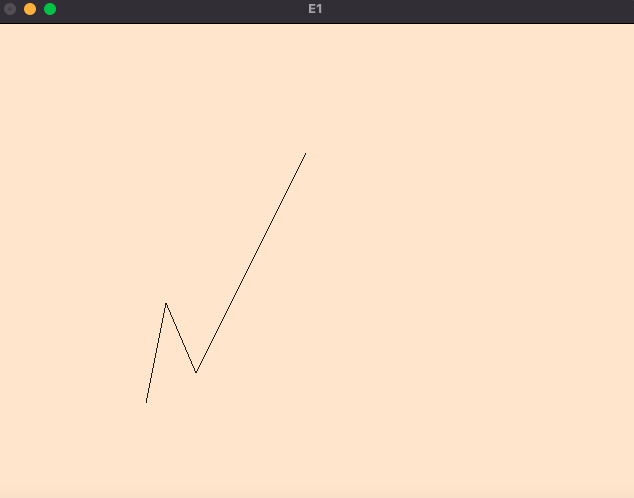
glutDisplayFunc(myDisplay);

myInit();

glutMainLoop();

**return** 1;

}

**

LINE\_LOOP:

#include<GLUT/glut.h>

**void** myInit() {

glClearColor(1.0,0.9,0.8,0.5);

glColor3f(0.1f,0.1f,0.1f);

glPointSize(5);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0,640.0,0.0,480.0);

}

**void** myDisplay() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_LINE\_LOOP);

glVertex2d(150,100);

glVertex2d(170,200);

glVertex2d(200,130);

glVertex2d(310,350);

glEnd();

glFlush();

}

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

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(640,480);

glutCreateWindow("E1");

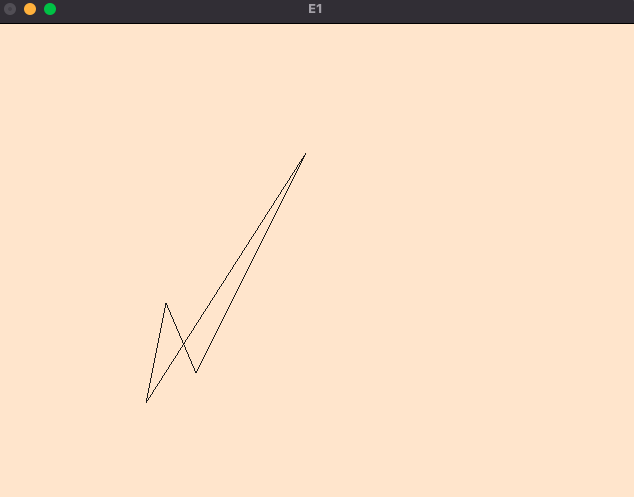
glutDisplayFunc(myDisplay);

myInit();

glutMainLoop();

**return** 1;

}

**

TRIANGLES:

#include<GLUT/glut.h>

**void** myInit() {

glClearColor(1.0,0.9,0.8,0.5);

glColor3f(0.1f,0.1f,0.1f);

glPointSize(5);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0,640.0,0.0,480.0);

}

**void** myDisplay() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_TRIANGLES);

glVertex2d(320,200);

glVertex2d(245,300);

glVertex2d(170,200);

glVertex2d(400,350);

glEnd();

glFlush();

}

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

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(640,480);

glutCreateWindow("E1");

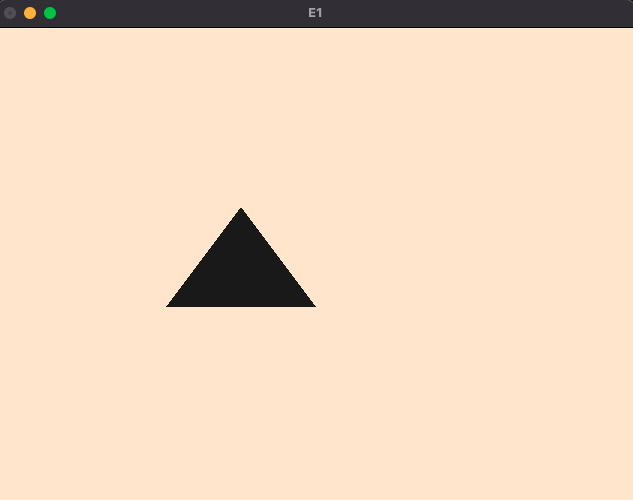
glutDisplayFunc(myDisplay);

myInit();

glutMainLoop();

**return** 1;

}

**

QUADS:

#include<GLUT/glut.h>

**void** myInit() {

glClearColor(1.0,0.9,0.8,0.5);

glColor3f(0.1f,0.1f,0.1f);

glPointSize(5);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0,640.0,0.0,480.0);

}

**void** myDisplay() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_QUADS);

glVertex2d(160,210);

glVertex2d(90,120);

glVertex2d(260,140);

glVertex2d(290,190);

glEnd();

glFlush();

}

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

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(640,480);

glutCreateWindow("E1");

glutDisplayFunc(myDisplay);

myInit();

glutMainLoop();

**return** 1;

}

QUAD\_STRIP:

#include<GLUT/glut.h>

**void** myInit() {

glClearColor(1.0,0.9,0.8,0.5);

glColor3f(0.1f,0.1f,0.1f);

glPointSize(5);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0,640.0,0.0,480.0);

}

**void** myDisplay() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_QUAD\_STRIP);

glVertex2d(160,210);

glVertex2d(90,120);

glVertex2d(260,140);

glVertex2d(290,190);

glEnd();

glFlush();

}

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

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(640,480);

glutCreateWindow("E1");

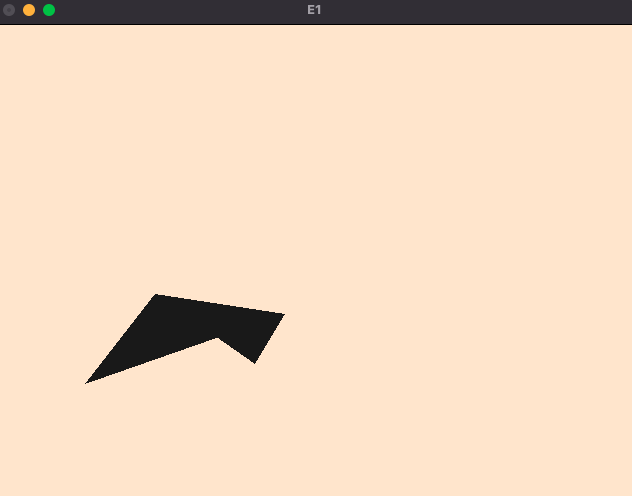
glutDisplayFunc(myDisplay);

myInit();

glutMainLoop();

**return** 1;

}

**

POLYGON:

#include<GLUT/glut.h>

**void** myInit() {

glClearColor(1.0,0.9,0.8,0.5);

glColor3f(0.1f,0.1f,0.1f);

glPointSize(5);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0,640.0,0.0,480.0);

}

**void** myDisplay() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_POLYGON);

glVertex2d(110,90);

glVertex2d(110,160);

glVertex2d(140,190);

glVertex2d(170,160);

glVertex2d(130,90);

glEnd();

glFlush();

}

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

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(640,480);

glutCreateWindow("E1");

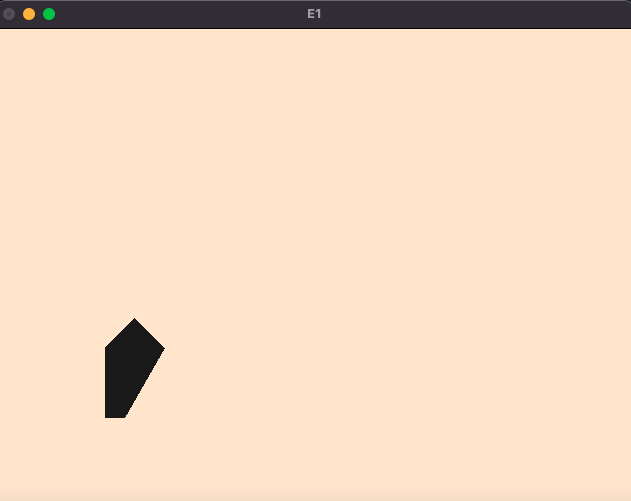
glutDisplayFunc(myDisplay);

myInit();

glutMainLoop();

**return** 1;

}

**

b) To create an output window and draw a checkerboard using OpenGL.

Source Code

#include<GLUT/glut.h>

**void** myInit() {

glClearColor(0.3,0.6,0.5,0.0);

glPointSize(10);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0,640.0,0.0,480.0);

}

**void** drawcheck(**float** x1,**float** y1,**float** x2, **float** y2,**float** x3,**float** y3,**float** x4,**float** y4,**int** color){

**if**(color==1){

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

}

**else**{

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

}

glBegin(GL\_QUADS);

glVertex2d(x1,y1);

glVertex2d(x2,y2);

glVertex2d(x3,y3);

glVertex2d(x4,y4);

glEnd();

}

**void** myDisplay() {

**int** color=1;

glClear(GL\_COLOR\_BUFFER\_BIT);

**for**(**int** i=50;i<=450;i+=50){

**for**(**int** j=50;j<=450;j+=50){

drawcheck(j+40,i+40,j+40,i+90,j+90,i+90,j+90,i+40,color);

**if**(color==1){

color=0;

}

**else**{

color=1;

}

}

}

glFlush(); }

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

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(640,480);

glutCreateWindow("E1 Checker board");

glutDisplayFunc(myDisplay);

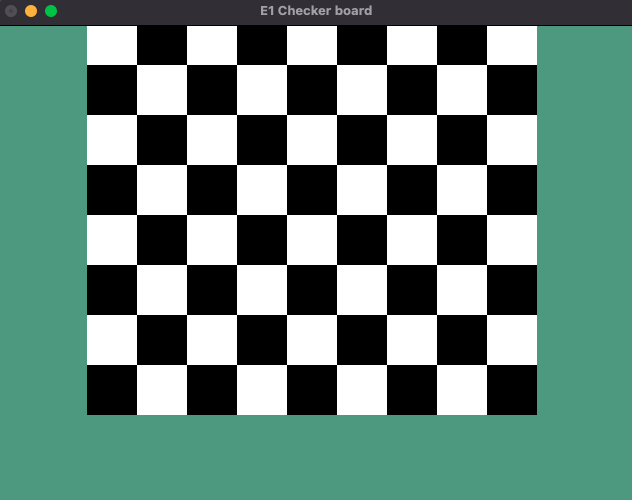
myInit();

glutMainLoop();

**return** 1;

}

Output



c) To create an output window and draw a house using POINTS,LINES,TRIANGLES and QUADS/POLYGON.

Source Code

#include<GLUT/glut.h>

#include<stdio.h>

**void** drawString(**float** x, **float** y, **const** **char** \*string) {

glRasterPos2f(x, y);

**for** (**const** **char** \*c = string; \*c != '\0'; c++) {

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_12, \*c);

}

}

**void** myInit() {

glClearColor(1.0, 1.0, 1.0, 0.0);

glPointSize(10);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

**void** myDisplay() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_QUADS);

glColor3f(0.57f, 0.74f, 0.80f);

glVertex2d(40, 40);

glVertex2d(240, 40);

glVertex2d(240, 240);

glVertex2d(40, 240);

glEnd();

glBegin(GL\_TRIANGLES);

glColor3f(0.3f, 0.2f, 0.1f);

glVertex2d(40, 240);

glVertex2d(140, 360);

glVertex2d(240, 240);

glEnd();

glBegin(GL\_QUADS);

glColor3f(0.4f, 0.3f, 0.2f);

glVertex2d(140, 360);

glVertex2d(450, 360);

glVertex2d(550, 240);

glVertex2d(240, 240);

glEnd();

glBegin(GL\_QUADS);

glColor3f(0.67f, 0.84f, 0.90f);

glVertex2d(240, 240);

glVertex2d(550, 240);

glVertex2d(550, 40);

glVertex2d(240, 40);

glEnd();

glBegin(GL\_QUADS);

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

glVertex2d(120, 40);

glVertex2d(120, 100);

glVertex2d(160, 100);

glVertex2d(160, 40);

glEnd();

glBegin(GL\_QUADS);

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

glVertex2d(300, 120);

glVertex2d(300, 160);

glVertex2d(340, 160);

glVertex2d(340, 120);

glEnd();

glBegin(GL\_LINES);

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

glVertex2d(320, 120);

glVertex2d(320, 160);

glVertex2d(300, 140);

glVertex2d(340, 140);

glEnd();

glBegin(GL\_QUADS);

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

glVertex2d(450, 120);

glVertex2d(450, 160);

glVertex2d(490, 160);

glVertex2d(490, 120);

glEnd();

glBegin(GL\_LINES);

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

glVertex2d(470, 120);

glVertex2d(470, 160);

glVertex2d(450, 140);

glVertex2d(490, 140);

glEnd();

glFlush();

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(640, 480);

glutCreateWindow("E1 house");

glutDisplayFunc(myDisplay);

myInit();

glutMainLoop();

**return** 1;

}

Output

