**CGA-LAB MANUAL**

**EXE-1A:**

Write an open GL program to specify points on a plane.

#include<GL/glut.h>

void pdisplay()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_POINTS);

glColor3f(1.0,1.0,1.0);

glVertex2f(150.0,180.0);

glColor3f(1.0,0.0,1.0);

glVertex2f(180.0,150.0);

glColor3f(0.0,1.0,1.0);

glVertex2f(250.0,250.0);

glEnd();

glFlush();

}

void init()

{

glClearColor(0.0,0.0,0.0,1.0);

gluOrtho2D(500.0,0.0,0.0,500.0);

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(500.0,500.0);

glutInitWindowPosition(0.0,0.0);

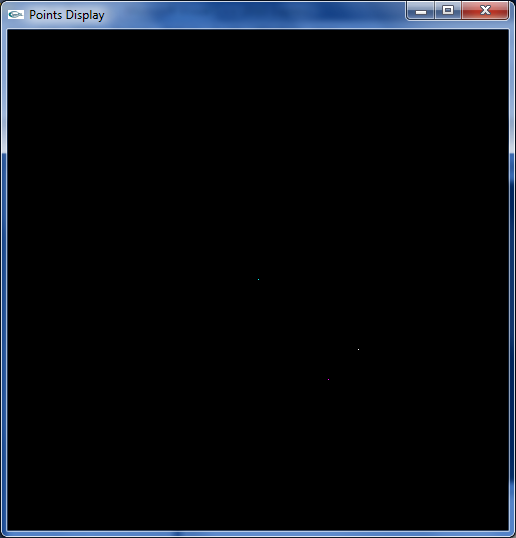
glutCreateWindow("Points Display");

glutDisplayFunc(pdisplay);

init();

glutMainLoop();

}



**EXE1B:**

Write an open GL to specify points large points on a plane.

#include<GL/glut.h>

void pdisplay()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_POINTS);

glColor3f(1.0,1.0,1.0);

glVertex2f(150.0,180.0);

glColor3f(1.0,0.0,1.0);

glVertex2f(180.0,150.0);

glColor3f(0.0,1.0,1.0);

glVertex2f(250.0,250.0);

glEnd();

glFlush();

}

void init()

{

glClearColor(0.0,0.0,0.0,1.0);

glPointSize(5.0);

gluOrtho2D(500.0,0.0,0.0,500.0);

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(500.0,500.0);

glutInitWindowPosition(0.0,0.0);

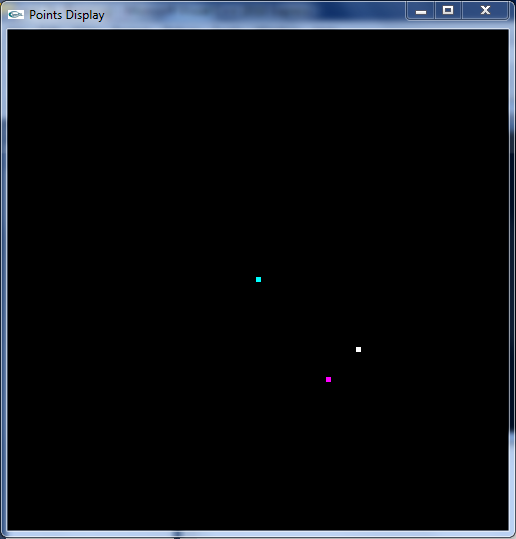
glutCreateWindow("Points Display");

glutDisplayFunc(pdisplay);

init();

glutMainLoop();

}



**Exe2a:**

Write an open GL program to draw lines on a plane.

#include<GL/glut.h>

void ldisplay()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_LINES);

glColor3f(1.0,0.0,0.0);

glVertex2f(250.0,250.0);

glVertex2f(150.0,150.0);

glColor3f(0.0,1.0,0.0);

glVertex2f(180.0,250.0);

glVertex2f(250.0,180.0);

glEnd();

glBegin(GL\_POINTS);

glColor3f(0.0,0.0,1.0);

glVertex2f(450.0,450.0);

glEnd();

glFlush();

}

void init()

{

glClearColor(1.0,1.0,1.0,1.0);

glPointSize(10.0);

gluOrtho2D(0.0,500.0,0.0,500.0);

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(500.0,500.0);

glutInitWindowPosition(100.0,100.0);

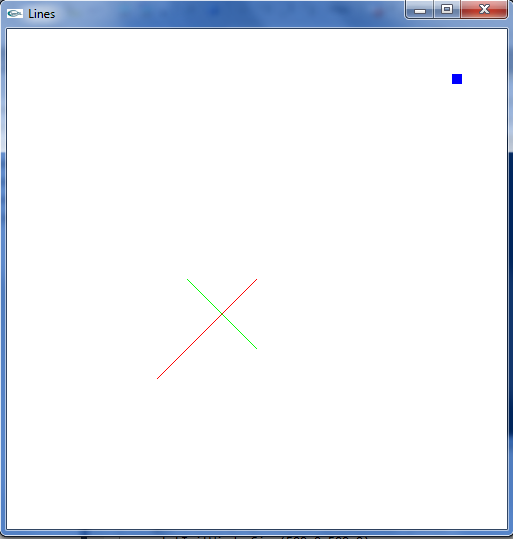
glutCreateWindow("Lines");

glutDisplayFunc(ldisplay);

init();

glutMainLoop();

}



**EXE2B:**

Write an openGL program to draw lines on a plane and increase the width .

#include<GL/glut.h>

void ldisplay()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_LINES);

glColor3f(1.0,0.0,0.0);

glVertex2f(250.0,250.0);

glVertex2f(150.0,150.0);

glColor3f(0.0,1.0,0.0);

glVertex2f(180.0,250.0);

glVertex2f(250.0,180.0);

glEnd();

glBegin(GL\_POINTS);

glColor3f(0.0,0.0,1.0);

glVertex2f(450.0,450.0);

glEnd();

glFlush();

}

void init()

{

glClearColor(1.0,1.0,1.0,1.0);

glPointSize(10.0);

glLineWidth(5.0);

gluOrtho2D(0.0,500.0,0.0,500.0);

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(500.0,500.0);

glutInitWindowPosition(100.0,100.0);

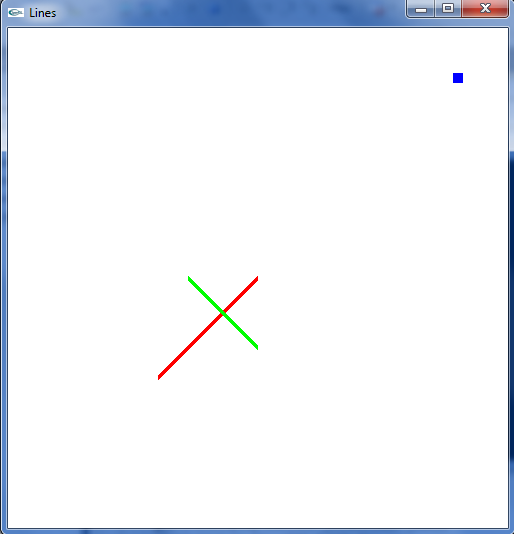
glutCreateWindow("Lines");

glutDisplayFunc(ldisplay);

init();

glutMainLoop();

}



**Exe4:**

Write an open GL program to draw a white rectangle on a black background.

#include<GL/glut.h>

void rdisplay()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0,1.0,1.0);

glBegin(GL\_POLYGON);

glVertex2f(100.0,100.0);

glVertex2f(150.0,100.0);

//glVertex2f(150.0,100.0);

glVertex2f(150.0,200.0);

//glVertex2f(150.0,200.0);

glVertex2f(100.0,200.0);

//glVertex2f(100.0,200.0);

//glVertex2f(100.0,100.0);

glEnd();

glFlush();

}

void init()

{

glClearColor(0.0,0.0,0.0,1.0);

glLineWidth(5.0);

gluOrtho2D(0.0,500.0,0.0,500.0);

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(500.0,500.0);

glutInitWindowPosition(0.0,0.0);

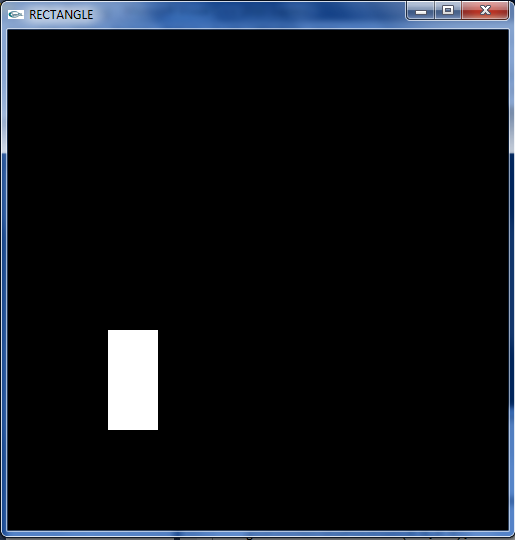
glutCreateWindow("RECTANGLE");

glutDisplayFunc(rdisplay);

init();

glutMainLoop();

}



**House using lines:**

Write an open GL program to draw a house using lines on a plane.

#include<GL/glut.h>

void housee()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_LINES);

glColor3f(0.0,0.0,0.0);

glVertex2f(100.0,500.0);

glVertex2f(100.0,100.0);

glVertex2f(100.0,100.0);

glVertex2f(500.0,100.0);

glVertex2f(500.0,100.0);

glVertex2f(500.0,500.0);

glVertex2f(500.0,500.0);

glVertex2f(100.0,500.0);

glVertex2f(500.0,500.0);

glVertex2f(300.0,700.0);

glVertex2f(300.0,700.0);

glVertex2f(100.0,500.0);

glVertex2f(200.0,100.0);

glVertex2f(200.0,300.0);

glVertex2f(200.0,300.0);

glVertex2f(400.0,300.0);

glVertex2f(400.0,300.0);

glVertex2f(400.0,100.0);

glEnd();

glFlush();

}

void init()

{

glClearColor(1.0,1.0,1.0,1.0);

glLineWidth(3.0);

gluOrtho2D(0.0,1000.0,0.0,1000.0);

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(1000.0,1000.0);

glutInitWindowPosition(0.0,0.0);

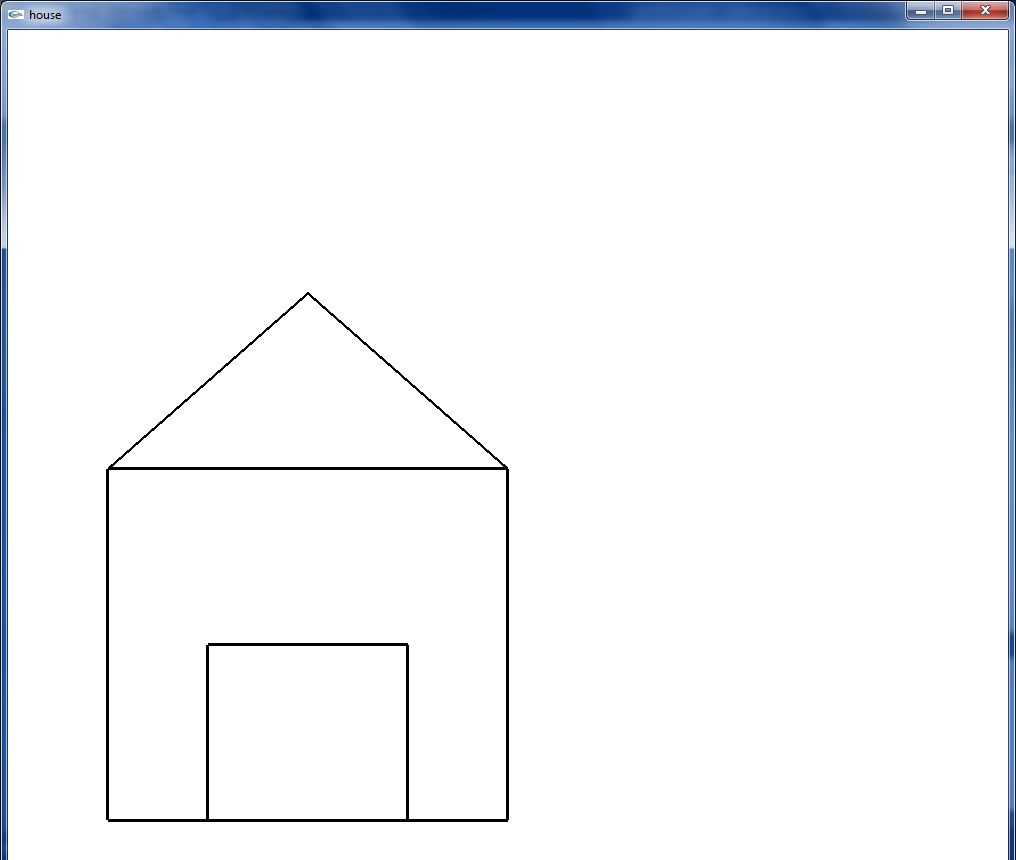
glutCreateWindow("house");

glutDisplayFunc(housee);

init();

glutMainLoop();

}



**House using polygons:**

Write an open GL program to draw house using polygons on a plane.

#include<GL/glut.h>

void housee()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_POLYGON);

glColor3f(1.0,1.0,0.0);

glVertex2f(100.0,500.0);

glVertex2f(100.0,100.0);

glVertex2f(500.0,100.0);

glVertex2f(500.0,500.0);

glEnd();

glBegin(GL\_TRIANGLES);

glColor3f(0.9,0.9,0.5);

glVertex2f(500.0,500.0);

glVertex2f(300.0,700.0);

glVertex2f(100.0,500.0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.3,0.3,0.5);

glVertex2f(200.0,100.0);

glVertex2f(200.0,300.0);

glVertex2f(400.0,300.0);

glVertex2f(400.0,100.0);

glEnd();

glFlush();

}

void init()

{

glClearColor(0.0,0.8,0.0,1.0);

glLineWidth(3.0);

gluOrtho2D(0.0,1000.0,0.0,1000.0);

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(1000.0,1000.0);

glutInitWindowPosition(0.0,0.0);

glutCreateWindow("house");

glutDisplayFunc(housee);

init();

glutMainLoop();

}



**Mouse and keyboard events:**

**Write an open gl program to generate mouse and keyboard events for closing a window.**

#include<GL/glut.h>

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_POINTS);

glColor3f(1.0,0.0,0.0);

glVertex2f(250.0,250.0);

glEnd();

glFlush();

}

void mice(int button, int state ,int x, int y)

{

if(button==GLUT\_LEFT\_BUTTON && state==GLUT\_DOWN)

{

exit(0);

}

}

void keyevent(unsigned char key, int x, int y)

{

if(key=='q'||key=='Q')

{

exit(0);

}

}

void init()

{

glClearColor(1.0,1.0,1.0,1.0);

glColor3f(1.0,0.0,0.0);

glPointSize(25.0);

gluOrtho2D(0.0,500.0,0.0,500.0);

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(500.0,500.0);

glutInitWindowPosition(0.0,0.0);

glutCreateWindow("events");

glutMouseFunc(mice);

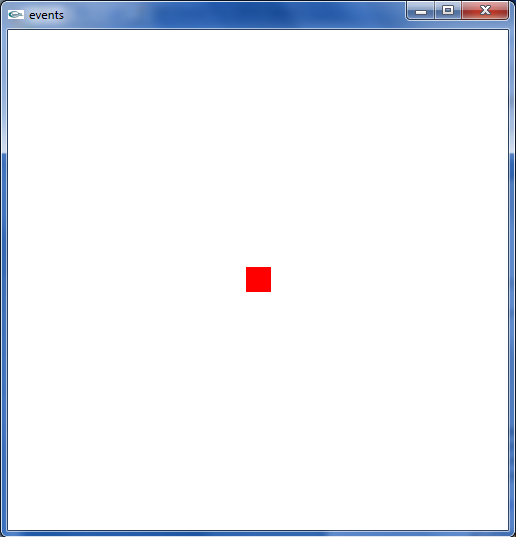
glutKeyboardFunc(keyevent);

glutDisplayFunc(display);

init();

glutMainLoop();

}



**Write an open gl program to generate a square when mouse is clicked.**

#include<GL/glut.h>

GLint x=50;

GLint y=50;

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glFlush();

}

void drawSquare(int x, int y)

{

glColor3f(1.0,0.0,0.0);

glBegin(GL\_POLYGON);

glVertex2f(x+100.0,y+100.0);

glVertex2f(x+100.0,y+150.0);

glVertex2f(x+150.0,y+150.0);

glVertex2f(x+150.0,y+100.0);

glEnd();

glFlush();

}

void mousee(int button,int state,int x,int y)

{

if(button==GLUT\_LEFT\_BUTTON && state==GLUT\_DOWN)

{

drawSquare(x,y);

}

}

void keys(unsigned char key,int x,int y)

{

if(key=='q'||key=='Q')

{

exit(0);

}

}

void init()

{

glClearColor(1.0,1.0,1.0,1.0);

glColor3f(1.0,0.0,0.0);

gluOrtho2D(0.0,550.0,0.0,550.0);

}

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

{

glutInit(&argc,argv);

glutInitWindowSize(550.0,550.0);

glutInitWindowPosition(0.0,0.0);

glutCreateWindow("Objects with events");

glutMouseFunc(mousee);

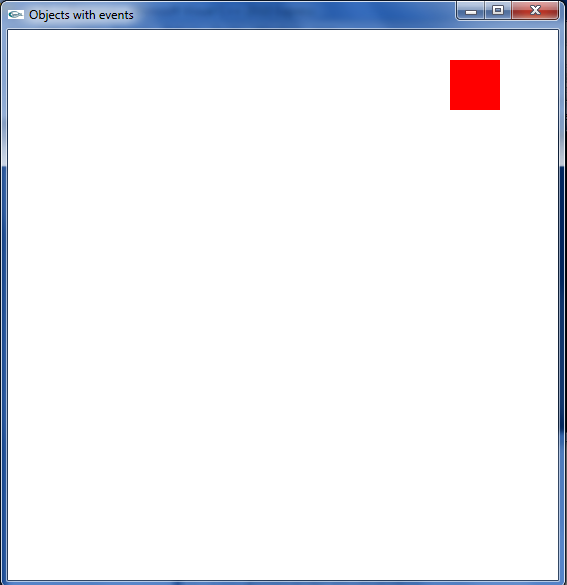
glutKeyboardFunc(keys);

glutDisplayFunc(display);

init();

glutMainLoop();

}

****