SSN COLLEGE OF ENGINEERING, KALAVAKKAM
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UCS1712 – GRAPHICS AND MULTIMEDIA LAB

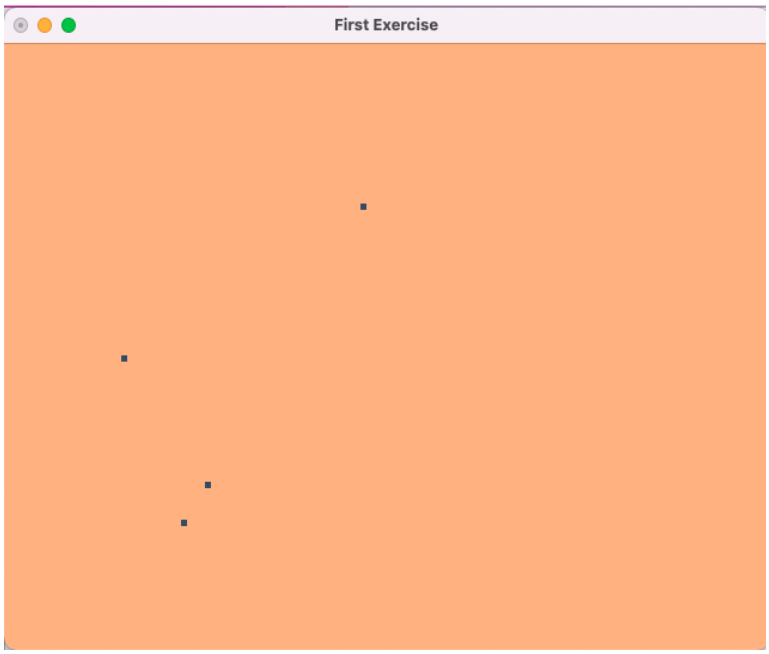
Lab Exercise 1 : Study of Basic Output Primitives in C++ using OpenGL

a). To create an output window using OPENGL and to draw the following basic output primitives– POINTS, LINES, LINE_STRIP, LINE_LOOP, TRIANGLES, QUADS, QUAD_STRIP, POLYGON.

Points:

```
#include<GLUT/glut.h>

void myInit() {

    glClearColor(1.0,0.7,0.5,0.0);

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

    glPointSize(5);

    glMatrixMode(GL_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(0.0,640.0,0.0,480.0);

}


void myDisplay() {

    glClear(GL_COLOR_BUFFER_BIT);

    glBegin(GL_POINTS);

    glVertex2d(150,100);

    glVertex2d(100,230);

    glVertex2d(170,130);

    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;
}
```

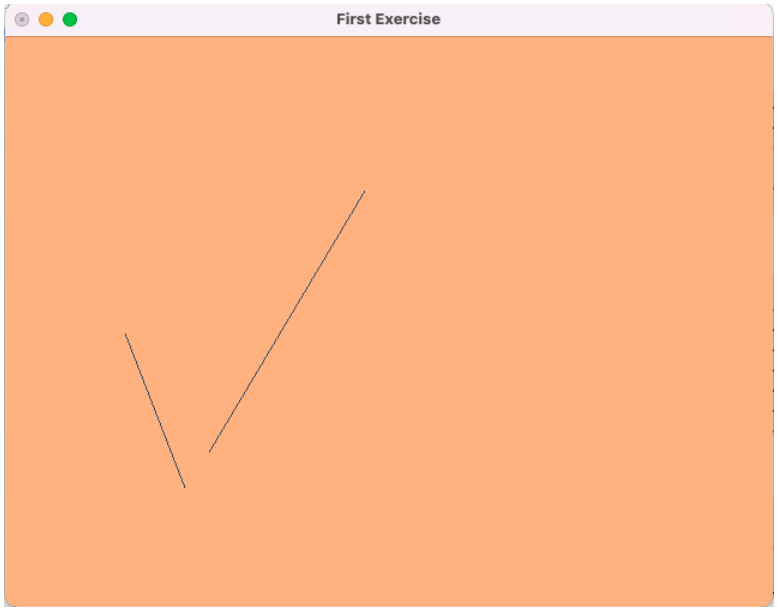


LINES:

```
#include<GLUT/glut.h>
void myInit() {
```

```
    glClearColor(1.0,0.7,0.5,0.0);

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

    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(100,230);
    glVertex2d(170,130);
    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;
```

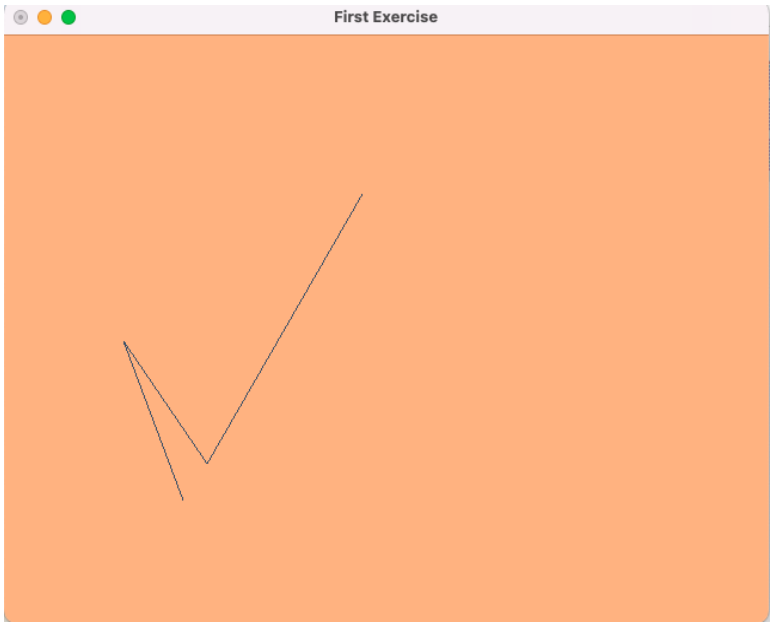
```
}
```



LINE_STRIP:

```
#include<GLUT/glut.h>

void myInit() {

    glClearColor(1.0,0.7,0.5,0.0);

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

    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(100,230);
    glVertex2d(170,130);
    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;
}
```

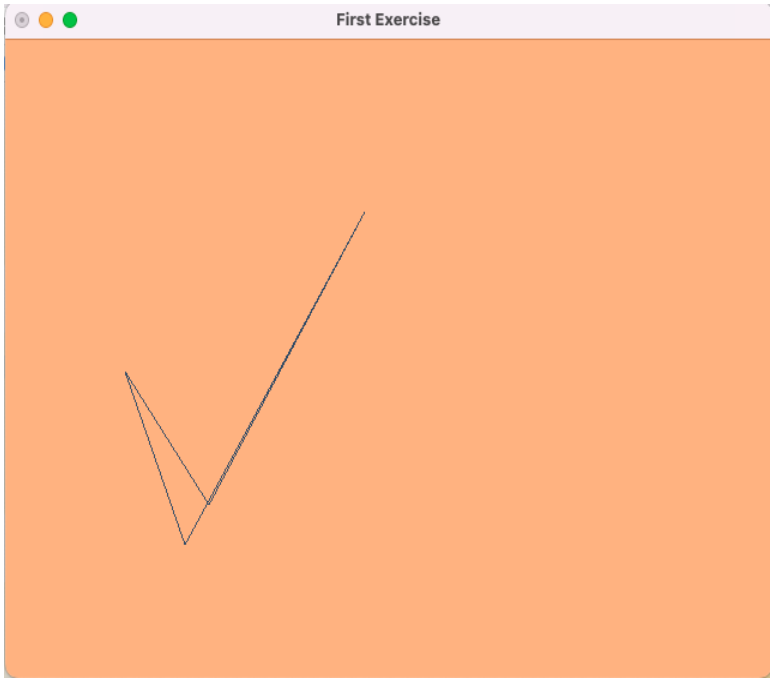


LINE_LOOP:

```
#include<GLUT/glut.h>

void myInit() {
```

```
    glClearColor(1.0,0.7,0.5,0.0);

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

    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(100,230);
    glVertex2d(170,130);
    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;

}
```

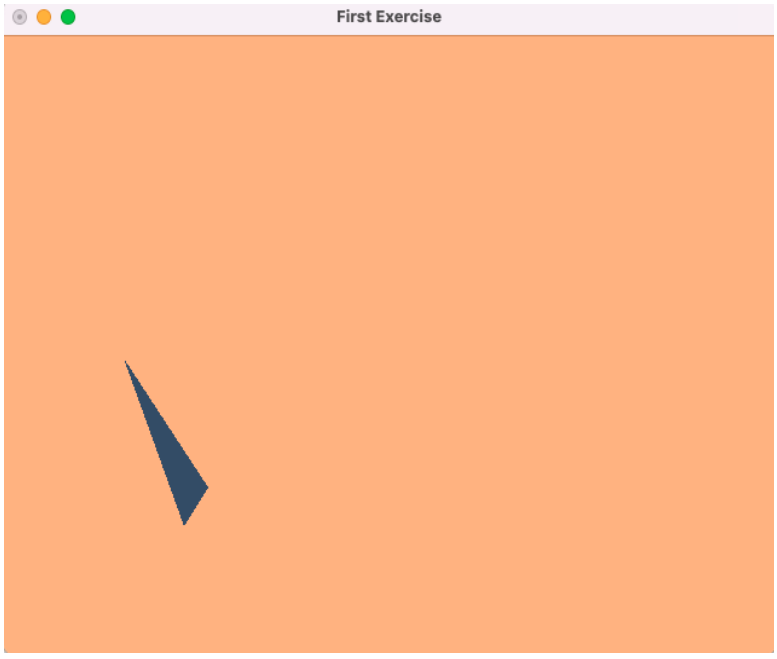
Name:Sanjay Rojar U *Register:205001089*



TRIANGLES:

```
#include<GLUT/glut.h>

void myInit() {

    glClearColor(1.0,0.7,0.5,0.0);

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

    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(150,100);
    glVertex2d(100,230);
    glVertex2d(170,130);
    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;

}
```

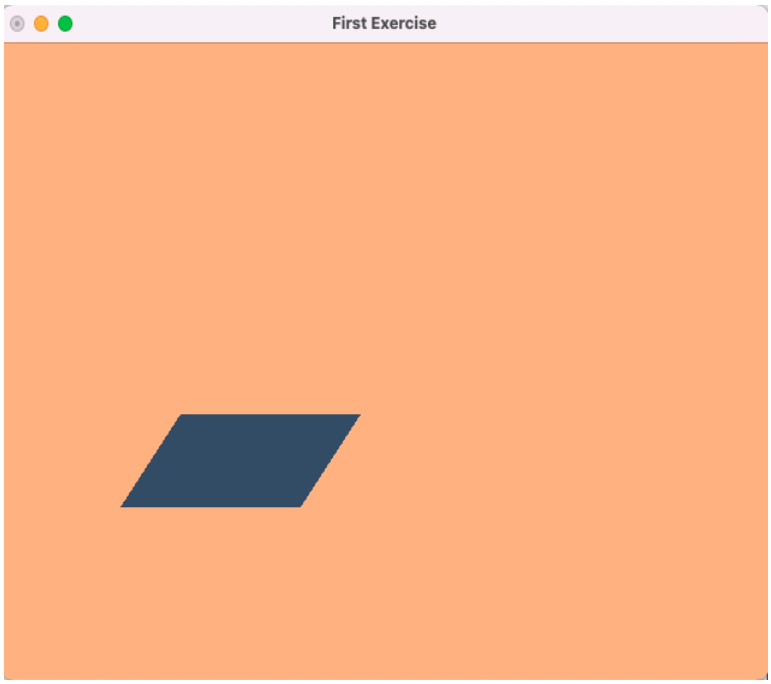


QUADS:

Name:Sanjay Rojar U *Register:205001089*

```
#include<GLUT/glut.h>

void myInit() {

    glClearColor(1.0,0.7,0.5,0.0);

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

    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(150,200);
    glVertex2d(100,130);
    glVertex2d(250,130);
    glVertex2d(300,200);
    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();
```

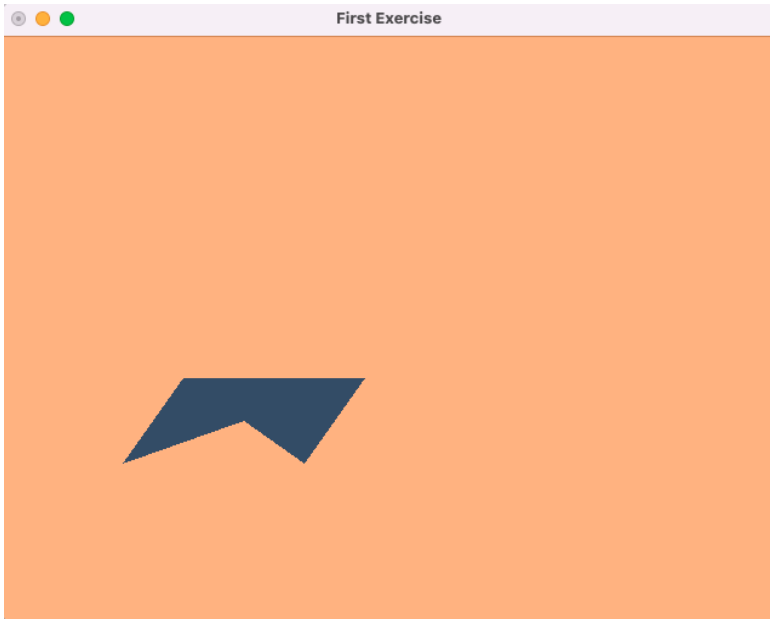
```
    return 1;
}
```



QUAD_STRIP:

```
#include<GLUT/glut.h>

void myInit() {

    glClearColor(1.0,0.7,0.5,0.0);

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

    glPointSize(5);

    glMatrixMode(GL_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(0.0,640.0,0.0,480.0);

}


void myDisplay() {
    glClear(GL_COLOR_BUFFER_BIT);
```

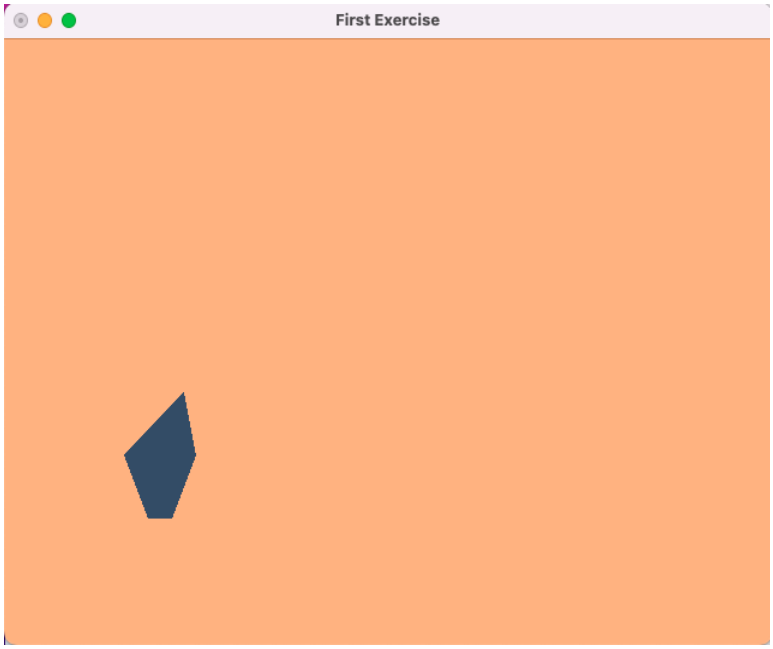
```
    glBegin(GL_QUAD_STRIP);
    glVertex2d(150,200);
    glVertex2d(100,130);
    glVertex2d(250,130);
    glVertex2d(300,200);
    glEnd();
    glFlush();
}



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;

}
```



POLYGON:

```
#include<GLUT/glut.h>

void myInit() {

    glClearColor(1.0,0.7,0.5,0.0);

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

    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(120,100);
    glVertex2d(100,150);
    glVertex2d(150,200);
    glVertex2d(160,150);
    glVertex2d(140,100);
    glEnd();
    glFlush();
}




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

    glutInit(&argc,argv);

    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);

    glutInitWindowSize(640,480);

    glutCreateWindow("First Exercise");
```

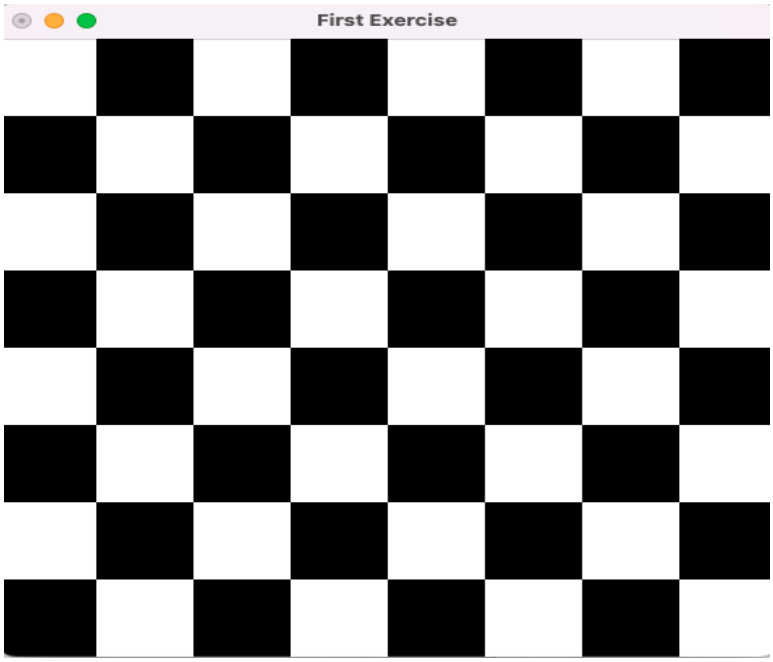
```
    glutDisplayFunc(myDisplay);

    myInit();

    glutMainLoop();

    return 1;

}
```



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

```
#include<GLUT/glut.h>
void myInit() {
    glClearColor(1.0,1.0,1.0,0.0);

    glPointSize(7);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0,480.0,0.0,480.0);
}
void drawsquare(int x1,int y1,int x2,int y2,int x3,int y3,int x4,int y4,int
clr)
{
    if(clr==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() {
    glClear(GL_COLOR_BUFFER_BIT);
    int clr=1;
    for(int i=0;i<480;i+=60)
    {
        for(int j=0;j<480;j+=60)
        {
            drawsquare(i, j, i, j+60, i+60, j+60, i+60, j,clr);
            if(clr==1)
                clr=0;
            else
                clr=1;
        }
        if(clr==1)
            clr=0;
        else
            clr=1;
    }
    glFlush();
}
int main(int argc,char* argv[]) {
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutInitWindowSize(480,480);
    glutCreateWindow("First Exercise");
    glutDisplayFunc(myDisplay);
    myInit();
    glutMainLoop();
    return 1;
}
```



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

```
#include<GLUT/glut.h>

void myInit() {

    glClearColor(1.0,1.0,1.0,0.0);

    glPointSize(5);

    glMatrixMode(GL_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(0.0,640.0,0.0,480.0);

}
```

```
void myDisplay() {
```

```
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0f,0.3f,0.8f);
    glBegin(GL_TRIANGLES);
    glVertex2i(120,240);
    glVertex2i(195,300);
    glVertex2i(270,240);
    glEnd();
    glColor3f(0.3f,0.3f,0.8f);
    glBegin(GL_QUADS);
    glVertex2i(120,100);
    glVertex2i(270,100);
    glVertex2i(270,240);
    glVertex2i(120,240);
    glEnd();
    glBegin(GL_QUADS);
    glVertex2i(195,300);
    glVertex2i(460,300);
    glVertex2i(520,240);
    glVertex2i(270,240);
    glEnd();
    glColor3f(0.0f,0.3f,0.8f);
    glBegin(GL_QUADS);
    glVertex2i(270,240);
    glVertex2i(520,240);
    glVertex2i(520,100);
    glVertex2i(270,100);
```

```
    glEnd();
    glBegin(GL_QUADS);
    glVertex2i(270,240);
    glVertex2i(520,240);
    glVertex2i(520,100);
    glVertex2i(270,100);
    glEnd();
    glColor3f(0.0f,0.0f,0.0f);
    glBegin(GL_QUADS);
    glVertex2i(270,240);
    glVertex2i(520,240);
    glVertex2i(520,100);
    glVertex2i(270,100);
    glEnd();
    glFlush();
}
```

```
int main(int argc,char* argv[]) {
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutInitWindowSize(640,480);
    glutCreateWindow("House");
    glutDisplayFunc(myDisplay);
```

```
    myInit();

    glutMainLoop();

    return 1;

}
```