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.

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

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

CODE:

```
#include<GL/glut.h>
void myInit()
{
        glClearColor(1.0, 1.0, 1.0, 1.0);
        glColor3f(0.0f, 0.0f, 0.0f);
        glPointSize(5);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        gluOrtho2D(0.0, 1440.0, 0.0, 480.0);
}
void myDispA() {
        glClear(GL_COLOR_BUFFER_BIT);
        glBegin(GL_POLYGON);
        glVertex2d(100, 100);
        glVertex2d(150, 230);
        glVertex2d(170, 130);
        glVertex2d(300, 350);
        glEnd();
        glFlush();
}
void myDispB()
{
        glClear(GL_COLOR_BUFFER_BIT);
        glBegin(GL_LINES);
        for (int i = 0; i <= 8; i++)
        {
                glVertex2d(120 + i * 50, 40);
                glVertex2d(120 + i * 50, 440);
        }
        for (int i = 0; i <= 8; i++)
        {
                glVertex2d(120 , 40 + i * 50);
                glVertex2d(520 , 40 + i*50);
        }
        glEnd();
        glBegin(GL_QUADS);
        for (int i = 0; i < 4; i++)
        {
                for (int j = 0; j < 4; j++)
                {
                        glVertex2d(120 + j * 100, 40 + i * 100);
                        glVertex2d(170 + j * 100, 40 + i * 100);
                        glVertex2d(170 + j * 100, 90 + i * 100);
                        glVertex2d(120 + j * 100, 90 + i * 100);
                }
        for (int j = 0; j < 4; j++)
        {
            glVertex2d(170 + j * 100, 90 + i * 100);
            glVertex2d(220 + j * 100, 90 + i * 100);
            glVertex2d(220 + j * 100, 140 + i * 100);
            glVertex2d(170 + j * 100, 140 + i * 100);
        }
    }
    glEnd();
    glFlush();
}
void myDispC() {
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_TRIANGLES);
    glVertex2d(320, 440);
    glVertex2d(120, 280);
    glVertex2d(520, 280);
    glEnd();
    glBegin(GL_LINE_STRIP);
    glVertex2d(170, 280);
    glVertex2d(170, 40);
    glVertex2d(470, 40);
    glVertex2d(470, 280);
    glEnd();
    glBegin(GL_QUADS);
    glVertex2d(220, 200);
    glVertex2d(300, 200);
    glVertex2d(300, 40);
    glVertex2d(220, 40);
    glEnd();
    glBegin(GL_LINE_LOOP);
    glVertex2d(360, 230);
    glVertex2d(420, 230);
    glVertex2d(420, 170);
    glVertex2d(360, 170);
    glEnd();
    glBegin(GL_QUADS);
    glVertex2d(360, 230);
    glVertex2d(375, 220);
    glVertex2d(375, 180);
    glVertex2d(360, 170);

    glVertex2d(420, 230);
    glVertex2d(405, 220);
    glVertex2d(405, 180);
    glVertex2d(420, 170);
    glEnd();
    glFlush();
}
void myDispD()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.45f, 0.45f, 0.45f);
    glBegin(GL_TRIANGLES);
    glVertex2d(120, 390);
    glVertex2d(430, 390);
    glVertex2d(275, 460);

    glVertex2d(1320, 210);
    glVertex2d(1170, 210);
    glVertex2d(1170, 290);
    glEnd();
```

```
glBegin(GL_QUADS);
glVertex2d(275, 390);
glVertex2d(275, 440);
glVertex2d(480, 440);
glVertex2d(480, 390);

glVertex2d(470, 210);
glVertex2d(470, 300);
glVertex2d(720, 300);
glVertex2d(720, 210);

glVertex2d(720, 210);
glVertex2d(720, 290);
glVertex2d(1170, 290);
glVertex2d(1170, 210);

glVertex2d(720, 200);
glVertex2d(720, 210);
glVertex2d(1320, 210);
glVertex2d(1320, 200);
glEnd();

glColor3f(1.0f, 0.9f, 0.6f);
glBegin(GL_QUADS);
glVertex2d(120, 40);
glVertex2d(120, 90);
glVertex2d(480, 90);
glVertex2d(480, 40);

glVertex2d(180, 90);
glVertex2d(180, 190);
glVertex2d(480, 190);
glVertex2d(480, 90);

glVertex2d(120, 190);
glVertex2d(120, 240);
glVertex2d(480, 240);
glVertex2d(480, 190);

glVertex2d(480, 210);
glVertex2d(720, 210);
glVertex2d(720, 40);
glVertex2d(480, 40);

glVertex2d(720, 45);
glVertex2d(1020, 45);
glVertex2d(1020, 210);
glVertex2d(720, 210);

glVertex2d(1020, 50);
glVertex2d(1020, 60);
glVertex2d(1320, 60);
glVertex2d(1320, 50);

glVertex2d(1020, 60);
glVertex2d(1020, 190);
glVertex2d(1280, 190);
glVertex2d(1280, 60);

glVertex2d(1020, 190);
glVertex2d(1020, 200);
glVertex2d(1320, 200);
```

```
glVertex2d(1320, 190);

glVertex2d(150, 240);
glVertex2d(150, 350);
glVertex2d(470, 350);
glVertex2d(470, 240);

glVertex2d(120, 350);
glVertex2d(120, 390);
glVertex2d(470, 390);
glVertex2d(470, 350);

glEnd();

glBegin(GL_TRIANGLES);
glVertex2d(720, 210);
glVertex2d(720, 300);
glVertex2d(1020, 210);
glEnd();

glColor3f(1.0f, 1.0f, 1.0f);
glBegin(GL_QUADS);
glVertex2d(210, 90);
glVertex2d(210, 180);
glVertex2d(240, 180);
glVertex2d(240, 90);

glVertex2d(250, 90);
glVertex2d(250, 180);
glVertex2d(280, 180);
glVertex2d(280, 90);

glVertex2d(290, 90);
glVertex2d(290, 180);
glVertex2d(320, 180);
glVertex2d(320, 90);

glVertex2d(330, 90);
glVertex2d(330, 180);
glVertex2d(360, 180);
glVertex2d(360, 90);

glVertex2d(490, 70);
glVertex2d(490, 180);
glVertex2d(650, 180);
glVertex2d(650, 70);

glVertex2d(655, 70);
glVertex2d(655, 180);
glVertex2d(680, 180);
glVertex2d(680, 70);

glVertex2d(690, 60);
glVertex2d(690, 190);
glVertex2d(700, 190);
glVertex2d(700, 60);

glVertex2d(750, 70);
glVertex2d(750, 180);
glVertex2d(990, 180);
glVertex2d(990, 70);
```

```
glVertex2d(1030, 80);
glVertex2d(1030, 170);
glVertex2d(1200, 170);
glVertex2d(1200, 80);

glVertex2d(170, 240);
glVertex2d(170, 340);
glVertex2d(410, 340);
glVertex2d(410, 240);

glEnd();

glColor3f(0.9f, 0.6f, 0.5f);
glBegin(GL_QUADS);
glVertex2d(1210, 80);
glVertex2d(1210, 170);
glVertex2d(1270, 170);
glVertex2d(1270, 80);
glEnd();

glColor3f(0.0f, 0.0f, 0.0f);
glBegin(GL_LINE_LOOP);
glVertex2d(120,40);
glVertex2d(120,90);
glVertex2d(180,90);
glVertex2d(180,190);
glVertex2d(120,190);
glVertex2d(120,240);
glVertex2d(480,240);
glVertex2d(480,210);
glVertex2d(720,210);
glVertex2d(720,190);
glVertex2d(480,190);
glVertex2d(480,60);
glVertex2d(720,60);
glVertex2d(720,40);
glEnd();
glBegin(GL_LINES);
glVertex2d(1020, 60);
glVertex2d(1320, 60);
glVertex2d(470, 300);
glVertex2d(720, 300);
glVertex2d(470, 390);
glVertex2d(480, 390);
glVertex2d(480, 390);
glVertex2d(480, 440);
glVertex2d(480, 440);
glVertex2d(318, 440);
glVertex2d(1170, 290);
glVertex2d(750, 290);
glEnd();
glBegin(GL_LINE_LOOP);
glVertex2d(210, 90);
glVertex2d(210, 180);
glVertex2d(240, 180);
glVertex2d(240, 90);
glEnd();
glBegin(GL_LINE_LOOP);
glVertex2d(250, 90);
glVertex2d(250, 180);
glVertex2d(280, 180);
glVertex2d(280, 90);
```

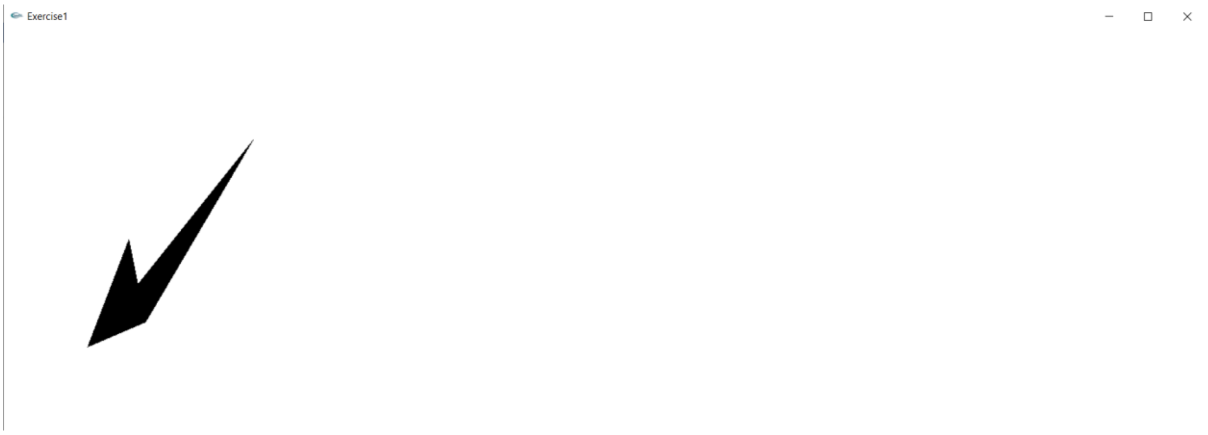
```
glEnd();
glBegin(GL_LINE_LOOP);
glVertex2d(290, 90);
glVertex2d(290, 180);
glVertex2d(320, 180);
glVertex2d(320, 90);
glEnd();
glBegin(GL_LINE_LOOP);
glVertex2d(330, 90);
glVertex2d(330, 180);
glVertex2d(360, 180);
glVertex2d(360, 90);
glEnd();
glBegin(GL_LINE_LOOP);
glVertex2d(490, 70);
glVertex2d(490, 180);
glVertex2d(650, 180);
glVertex2d(650, 70);
glEnd();
glBegin(GL_LINE_LOOP);
glVertex2d(655, 70);
glVertex2d(655, 180);
glVertex2d(680, 180);
glVertex2d(680, 70);
glEnd();
glBegin(GL_LINE_LOOP);
glVertex2d(690, 60);
glVertex2d(690, 190);
glVertex2d(700, 190);
glVertex2d(700, 60);
glEnd();
glBegin(GL_LINE_LOOP);
glVertex2d(750, 70);
glVertex2d(750, 180);
glVertex2d(990, 180);
glVertex2d(990, 70);
glEnd();
glBegin(GL_LINE_LOOP);
glVertex2d(1030, 80);
glVertex2d(1030, 170);
glVertex2d(1200, 170);
glVertex2d(1200, 80);
glEnd();
glBegin(GL_LINE_LOOP);
glVertex2d(170, 240);
glVertex2d(170, 340);
glVertex2d(410, 340);
glVertex2d(410, 240);
glEnd();
glBegin(GL_LINE_LOOP);
glVertex2d(1210, 80);
glVertex2d(1210, 170);
glVertex2d(1270, 170);
glVertex2d(1270, 80);
glEnd();
glBegin(GL_LINE_LOOP);
glVertex2d(150,240);
glVertex2d(150,350);
glVertex2d(430,350);
glVertex2d(430,240);
glVertex2d(410,240);
glVertex2d(410,340);
```

```
glVertex2d(170,340);
glVertex2d(170,240);
glEnd();
glBegin(GL_LINE_LOOP);
glVertex2d(120,350);
glVertex2d(120,390);
glVertex2d(470,390);
glVertex2d(470,240);
glVertex2d(430,240);
glVertex2d(430,350);
glEnd();
glBegin(GL_LINE_LOOP);
glVertex2d(720, 45);
glVertex2d(720, 300);
glVertex2d(1020, 210);
glVertex2d(1020, 45);
glEnd();
glBegin(GL_LINE_LOOP);
glVertex2d(1020, 200);
glVertex2d(1320, 200);
glVertex2d(1320, 190);
glVertex2d(1280, 190);
glVertex2d(1280, 60);
glVertex2d(1320, 60);
glVertex2d(1320, 50);
glVertex2d(1020, 50);
glEnd();
glBegin(GL_LINE_LOOP);
glVertex2d(120, 390);
glVertex2d(430, 390);
glVertex2d(275, 460);
glEnd();
glBegin(GL_LINE_LOOP);
glVertex2d(1320, 210);
glVertex2d(1020, 210);
glVertex2d(1170, 290);
glEnd();
glBegin(GL_QUADS);
glVertex2d(120, 390);
glVertex2d(130, 390);
glVertex2d(275, 455);
glVertex2d(275, 460);

glVertex2d(430, 390);
glVertex2d(420, 390);
glVertex2d(275, 455);
glVertex2d(275, 460);

glVertex2d(1020, 210);
glVertex2d(1030, 210);
glVertex2d(1170, 285);
glVertex2d(1170, 290);

glVertex2d(1320, 210);
glVertex2d(1310, 210);
glVertex2d(1170, 285);
glVertex2d(1170, 290);

glVertex2d(130, 240);
glVertex2d(135, 240);
glVertex2d(135, 350);
glVertex2d(130, 350);
```

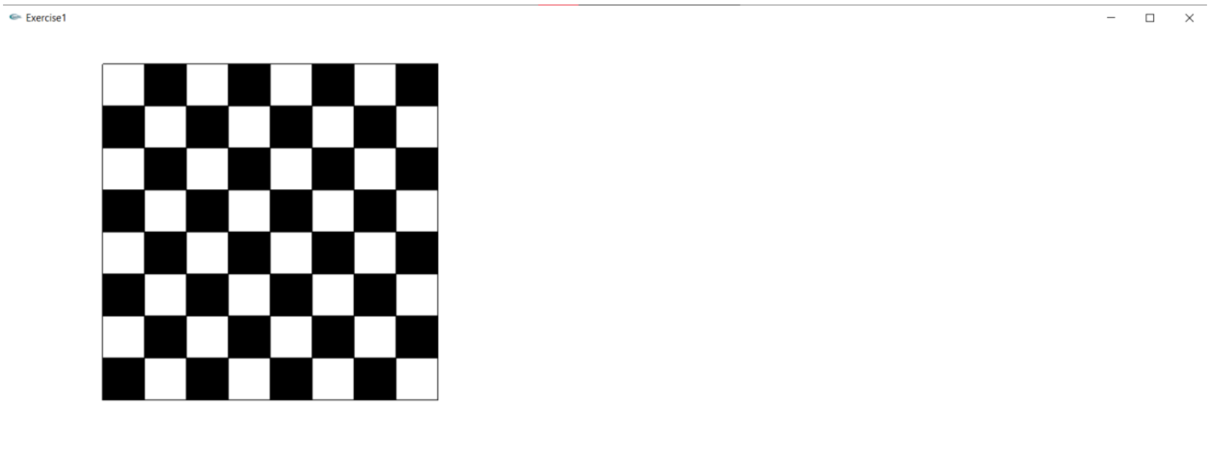
```
        glVertex2d(430, 240);
        glVertex2d(425, 240);
        glVertex2d(425, 350);
        glVertex2d(430, 350);

        glVertex2d(130, 270);
        glVertex2d(130, 267);
        glVertex2d(430, 267);
        glVertex2d(430, 270);

        for (int i = 1; i < 10; i++)
        {
               glVertex2d(130 + i * 30, 240);
               glVertex2d(132 + i * 30, 240);
               glVertex2d(132 + i * 30, 270);
               glVertex2d(130 + i * 30, 270);
        }

        glVertex2d(1290, 60);
        glVertex2d(1290, 190);
        glVertex2d(1292, 190);
        glVertex2d(1292, 60);

        glVertex2d(1310, 60);
        glVertex2d(1310, 190);
        glVertex2d(1312, 190);
        glVertex2d(1312, 60);

        glVertex2d(140, 90);
        glVertex2d(140, 190);
        glVertex2d(142, 190);
        glVertex2d(142, 90);

        glVertex2d(160, 90);
        glVertex2d(160, 190);
        glVertex2d(162, 190);
        glVertex2d(162, 90);
        glEnd();
        glBegin(GL_POINTS);
        glVertex2d(1220, 120);
        glEnd();
        glFlush();
}
int main(int argc, char* argv[])
{
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
        glutInitWindowSize(1440, 480);
        glutCreateWindow("Exercise1");
        glutDisplayFunc(myDispD); //To be changed for each function
        myInit();
        glutMainLoop();
        return 1;
}
```

Output:

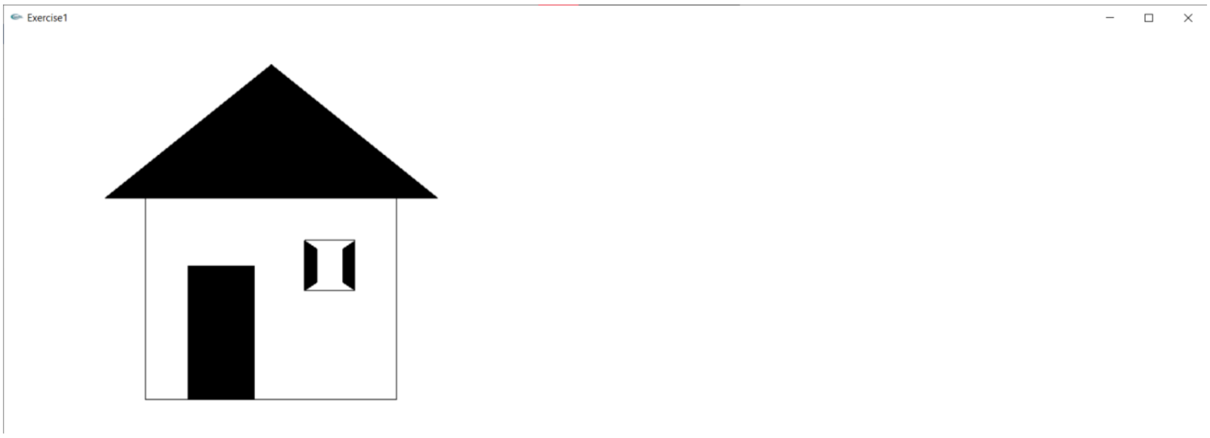
Polygon:

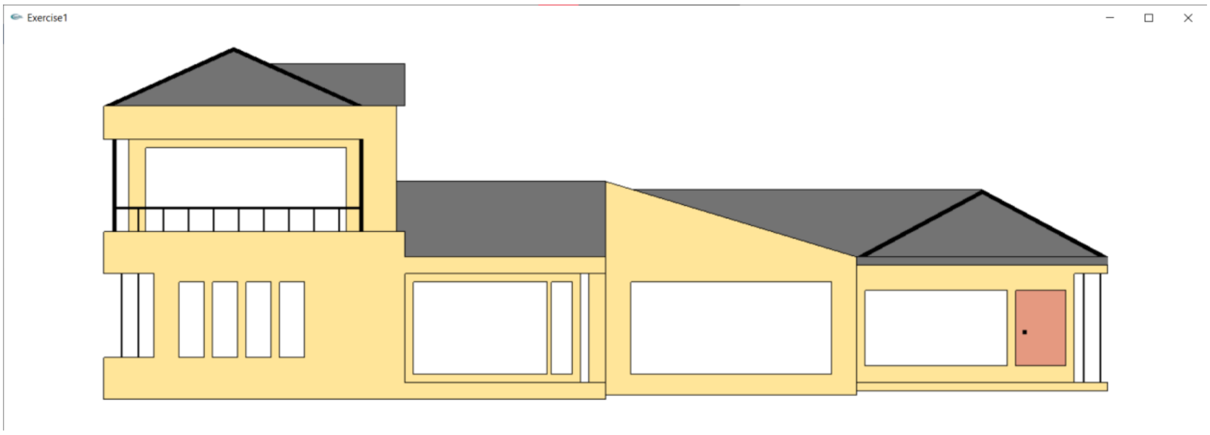


Checker Board:



House:





Result:

Thus Basic Output primitives have been studied and executed.