**UCS 1712 – GRAPHICS AND MULTIMEDIA LAB**

**ASSIGNMENT – 1**

**VISHAL N**

**185001198**

**27.07.2021 CSEC**

**1.**

**A)**

#include <GL/glut.h>

#include <GL/gl.h>

void myInit() {

    glClearColor(2.0, 1.0, 2.0, 0.0);

    glColor3f(0.0f, 0.0f, 0.0f);

    glPointSize(10);

    glMatrixMode(GL\_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(0.0, 800.0, 0.0, 600.0);

}

void basicPrimitives() {

    glClear(GL\_COLOR\_BUFFER\_BIT);

    glBegin(GL\_POINTS);

    glVertex2d(50, 550);

    glVertex2d(80, 540);

    glEnd();

    glBegin(GL\_LINES);

    glVertex2d(20, 450);

    glVertex2d(150, 450);

    glEnd();

    glBegin(GL\_LINE\_STRIP);

    glVertex2d(20, 350);

    glVertex2d(150, 350);

    glVertex2d(200, 250);

    glVertex2d(100, 250);

    glEnd();

    glBegin(GL\_LINE\_LOOP);

    glVertex2d(50, 150);

    glVertex2d(150, 150);

    glVertex2d(100, 50);

    glVertex2d(75, 50);

    glVertex2d(50, 150);

    glEnd();

    glBegin(GL\_TRIANGLES);

    glVertex2d(300, 550);

    glVertex2d(370, 480);

    glVertex2d(230, 480);

    glEnd();

    glBegin(GL\_QUADS);

    glVertex2d(320, 400);

    glVertex2d(450, 300);

    glVertex2d(420, 200);

    glVertex2d(300, 190);

    glEnd();

    glBegin(GL\_QUAD\_STRIP);

    glVertex2d(500, 550);

    glVertex2d(500, 450);

    glVertex2d(600, 550);

    glVertex2d(600, 450);

    glVertex2d(700, 450);

    glVertex2d(700, 350);

    glEnd();

    glBegin(GL\_POLYGON);

    glVertex2d(500, 250);

    glVertex2d(600, 250);

    glVertex2d(700, 150);

    glVertex2d(600, 150);

    glVertex2d(500, 150);

    glEnd();

    glFlush();

}

int main(int argc, char\* argv[]) {

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    glutInitWindowSize(800, 600);

    glutCreateWindow("Basic Output Primitives");

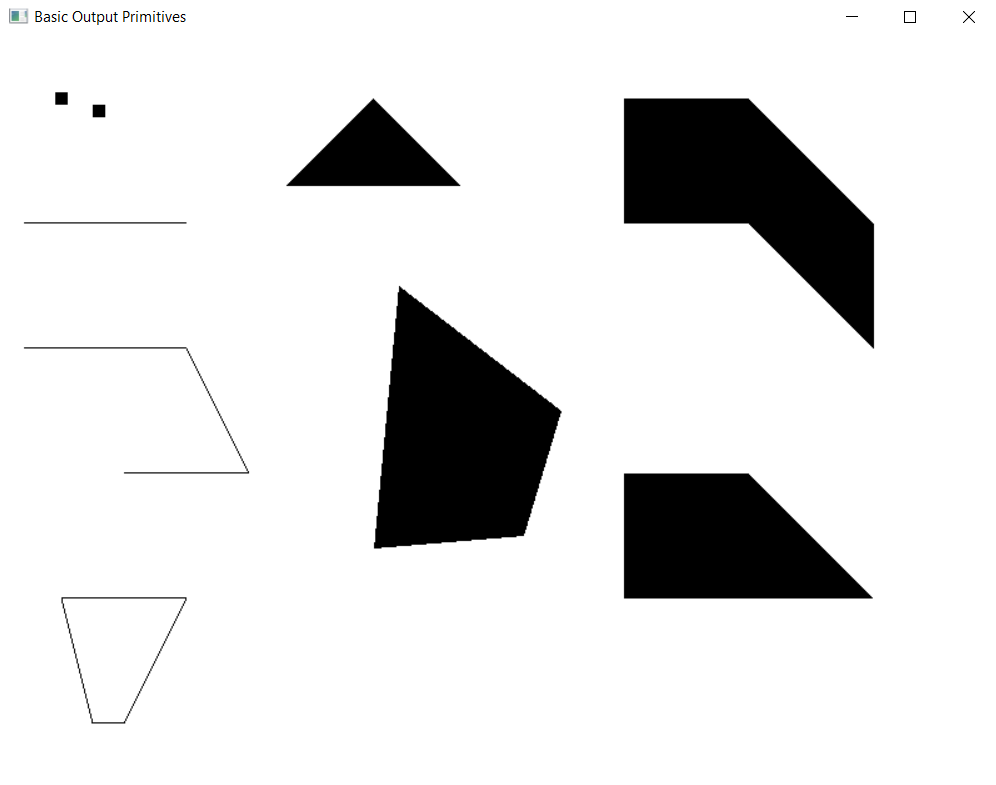
    glutDisplayFunc(basicPrimitives);

    myInit();

    glutMainLoop();

    return 1;

}

****

**B)**

#include <GL/glut.h>

#include <GL/gl.h>

#include <stdio.h>

void myInit() {

    glClearColor(2.0, 1.0, 2.0, 0.0);

    glColor3f(0.0f, 0.0f, 0.0f);

    glPointSize(10);

    glMatrixMode(GL\_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(0.0, 500.0, 0.0, 500.0);

}

void checker() {

    glClear(GL\_COLOR\_BUFFER\_BIT);

    int i, j;

    for (i = 0; i < 800; i += 100) {

        for (j = 0; j < 800; j += 50) {

            if (j % 100 == 0) {

                glBegin(GL\_QUADS);

                glVertex2d(i, j);

                glVertex2d(i, j + 50);

                glVertex2d(i + 50, j + 50);

                glVertex2d(i + 50, j);

                glEnd();

            }

            else {

                glBegin(GL\_QUADS);

                glVertex2d(i + 50, j);

                glVertex2d(i + 50, j + 50);

                glVertex2d(i + 100, j + 50);

                glVertex2d(i + 100, j);

                glEnd();

            }

        }

    }

    glFlush();

}

int main(int argc, char\* argv[]) {

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    glutInitWindowSize(500, 500);

    glutCreateWindow("Checker Pattern");

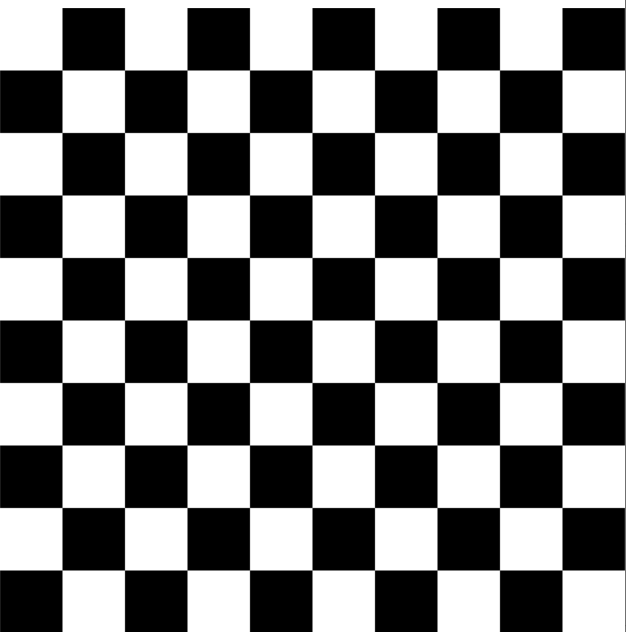
    glutDisplayFunc(checker);

    myInit();

    glutMainLoop();

    return 1;

}

****

**C)**

#include <GL/glut.h>

#include <GL/gl.h>

void myInit() {

    glClearColor(2.0, 1.0, 2.0, 0.0);

    glColor3f(0.0f, 0.0f, 0.0f);

    glPointSize(10);

    glMatrixMode(GL\_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

void house() {

    glClear(GL\_COLOR\_BUFFER\_BIT);

    glBegin(GL\_TRIANGLES);

    glColor3f(0.5, 0.0, 0.0);

    glVertex2d(200, 400);

    glVertex2d(150, 300);

    glVertex2d(250, 300);

    glEnd();

    glBegin(GL\_QUADS);

    glColor3f(0.0, 0.0, 0.5);

    glVertex2d(150, 300);

    glVertex2d(250, 300);

    glVertex2d(250, 200);

    glVertex2d(150, 200);

    glEnd();

    glBegin(GL\_QUADS);

    glColor3f(1.0, 0.0, 0.0);

    glVertex2d(200, 400);

    glVertex2d(400, 400);

    glVertex2d(450, 300);

    glVertex2d(250, 300);

    glEnd();

    glBegin(GL\_QUADS);

    glColor3f(0.0, 0.0, 1.0);

    glVertex2d(250, 300);

    glVertex2d(450, 300);

    glVertex2d(450, 200);

    glVertex2d(250, 200);

    glEnd();

    glFlush();

}

int main(int argc, char\* argv[]) {

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    glutInitWindowSize(640, 480);

    glutCreateWindow("House");

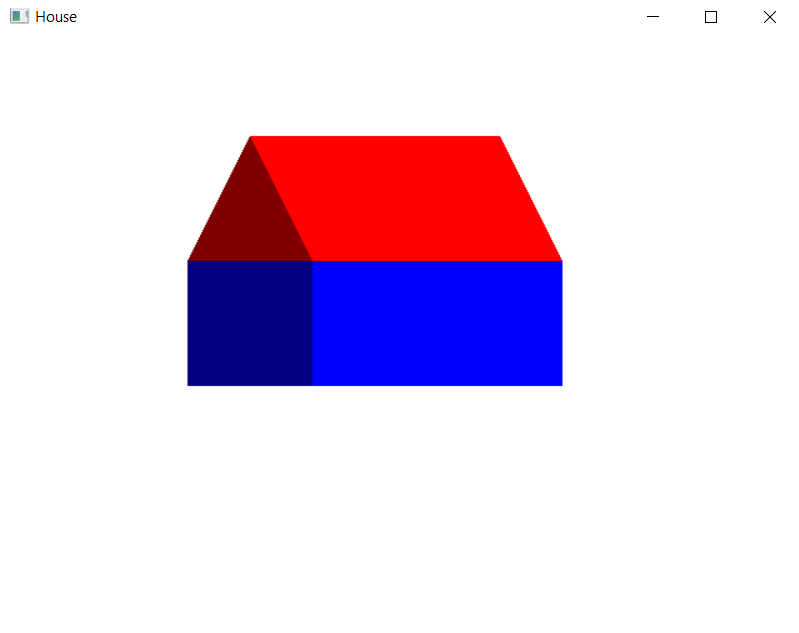
    glutDisplayFunc(house);

    myInit();

    glutMainLoop();

    return 1;

}

****