GRAPHICS LAB
EX 1

205001085
SABARIVASAN V

AIM:

Study of basic output primitives in c++ using openGL.

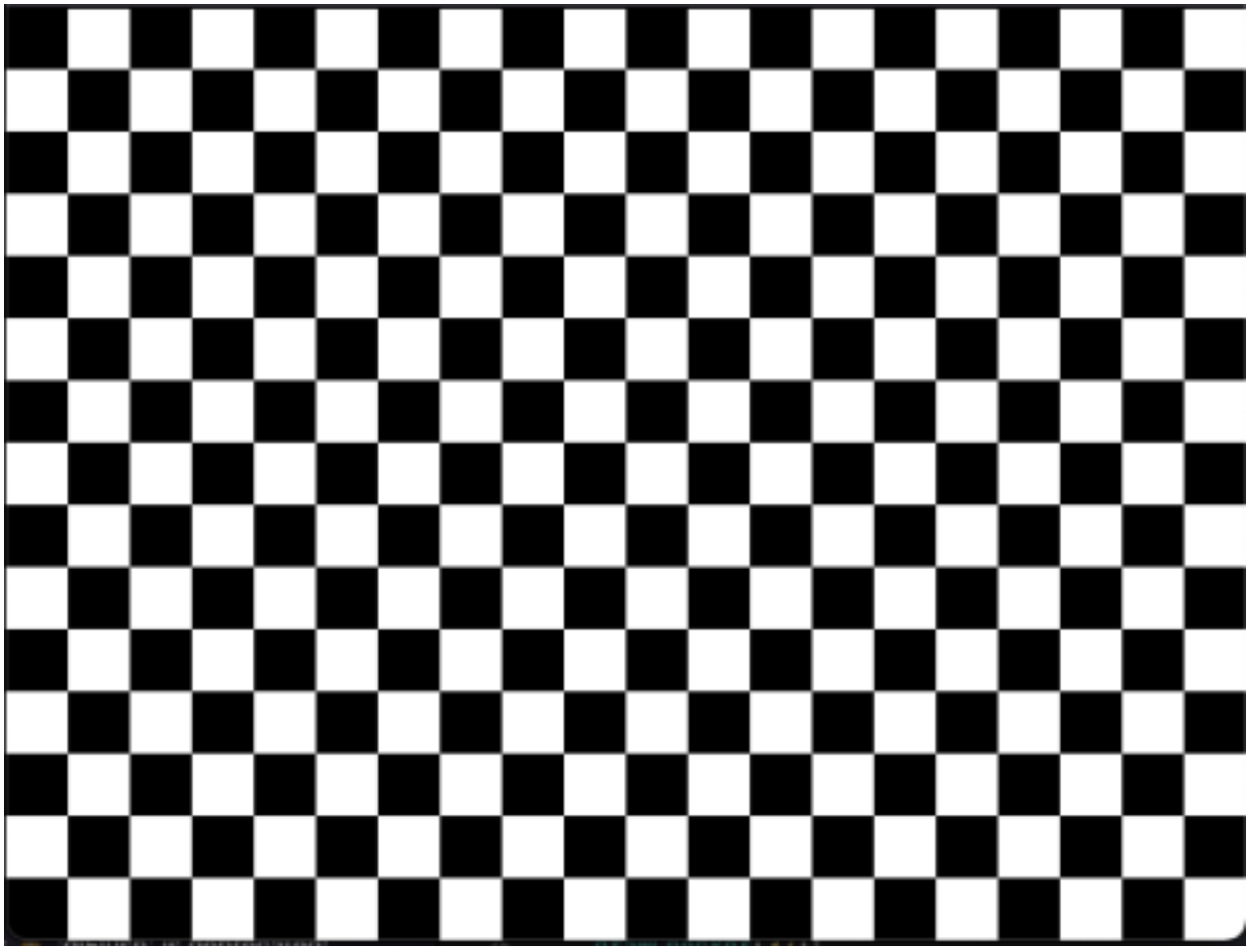
CODE:

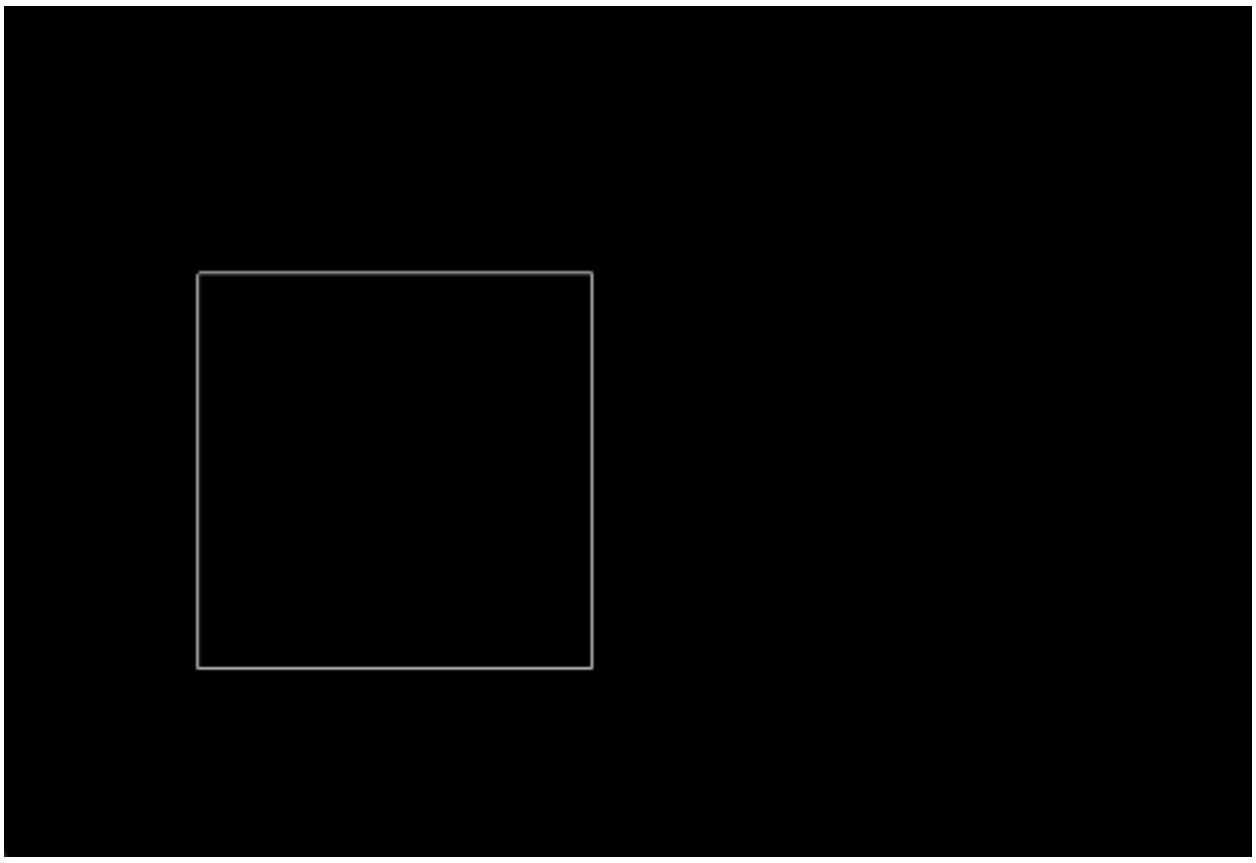
```
#include<GLUT/glut.h>
void myInit() {
glClearColor(0.0,0.0,0.0,0.0);
glColor3f(1.0f,1.0f,1.0f);
glPointSize(8);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluOrtho2D(0.0,640.0,0.0,480.0);
// glClearColor(1.0,1.0,1.0,0.0); // set white background color
// glMatrixMode(GL_PROJECTION);
// glLoadIdentity();
// gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}
void drawChecker(int size)
{
int i=0;
int j=0;
for (i = 0; i < 100 ; ++i) {
for (j = 0; j < 100; ++j) {
if((i + j)%2 == 0) // if i + j is even
glColor3f( 0.0, 0.0, 0.0);
else
glColor3f( 1.0, 1.0, 1.0);
glRecti(i*size, j*size, (i+1)*size, (j+1)*size); // draw the rectangle
}
}
glFlush();
}
void checkerboard(void) {
glClear(GL_COLOR_BUFFER_BIT); // clear the screen
drawChecker(32);
}
void myDisplay() {
glClear(GL_COLOR_BUFFER_BIT);
//glBegin(GL_LINE_STRIP);
glBegin(GL_LINE_LOOP);
//glBegin(GL_QUADS);
//glBegin(GL_QUAD_STRIP);
//glBegin(GL_POLYGON);
//glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, 'A'); glRasterPos2f(300, 100);
```

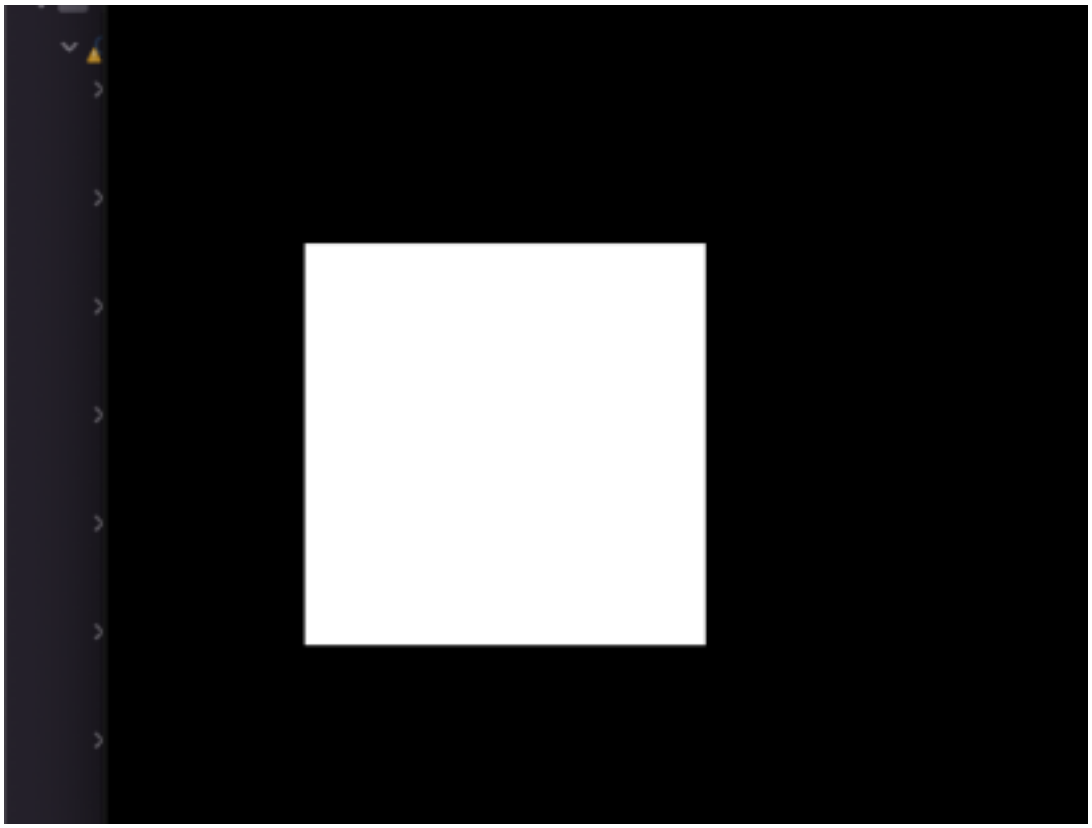
```
glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, 'A'); glVertex2d(300,100);
glVertex2d(100,100);
glVertex2d(100,300);
glVertex2d(300,300);
glEnd();
glBegin(GL_LINE_LOOP);
glVertex2d(100,300);
glVertex2d(300,300);
glVertex2d(200, 420);
glEnd();
// glBegin(GL_QUAD_STRIP);
// glVertex2d(170,100);
// glVertex2d(230,100);
// glVertex2d(170,170);
// glVertex2d(230,170);
// glEnd();
glBegin(GL_LINES);
glVertex2d(200,420);
glVertex2d(360,420);
glEnd();
glBegin(GL_LINES);
glVertex2d(300,300);
glVertex2d(460,300);
glEnd();
glBegin(GL_LINES);
glVertex2d(360,420);
glVertex2d(460,300);
glEnd();
glBegin(GL_LINES);
glVertex2d(460,300);
glVertex2d(460,100);
glEnd();
glBegin(GL_LINES);
glVertex2d(460,100);
glVertex2d(300,100);
glEnd();
glBegin(GL_LINES);
glVertex2d(170,100);
glVertex2d(170,170);
glEnd();
glBegin(GL_LINES);
glVertex2d(230,100);
glVertex2d(230,170);
glEnd();
glBegin(GL_LINES);
glVertex2d(170,170);
glVertex2d(230,170);
glEnd();
// glRasterPos2f(100, 150);
// glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18, 'A'); glFlush();
glBegin;
glEnd();
}
```

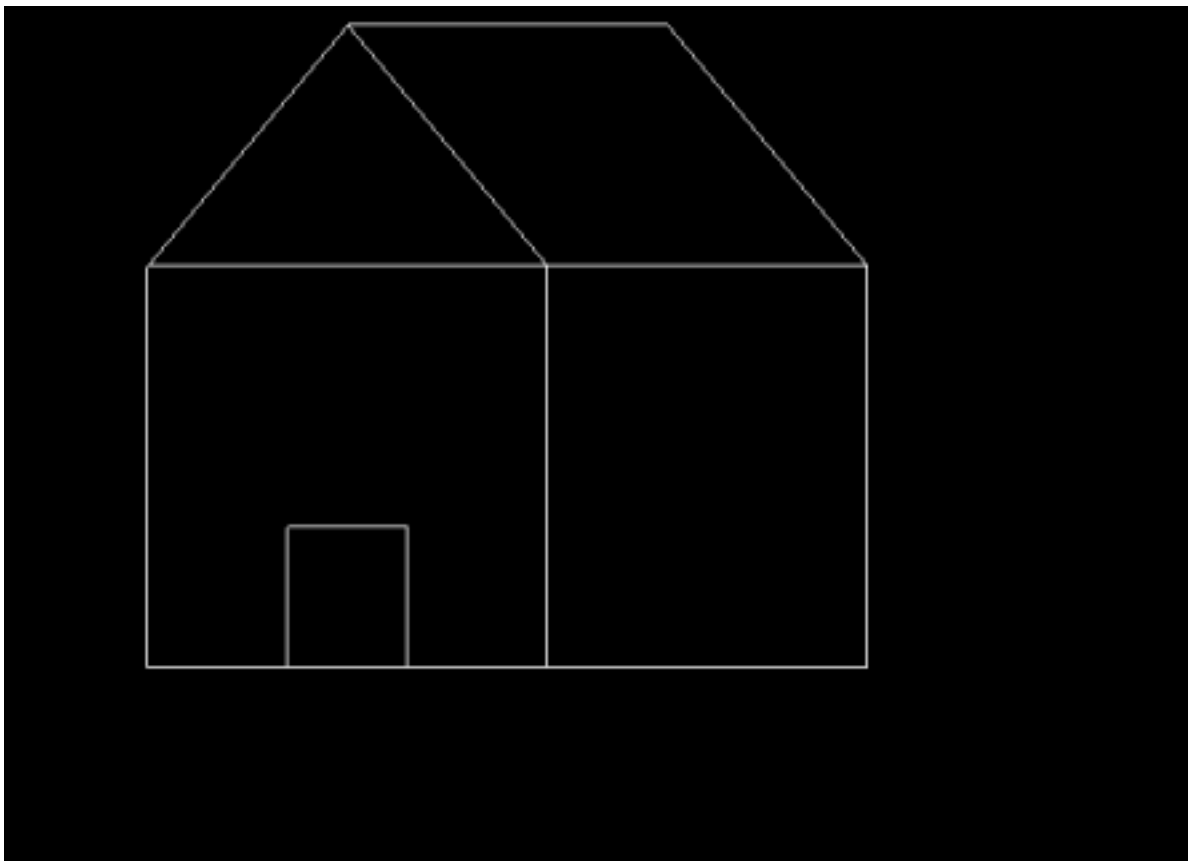
```
int main(int argc,char* argv[]) {
// glutInit(&argc,argv);
// glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
// glutInitWindowSize(640,480);
// glutCreateWindow("First Exercise");
// glutDisplayFunc(checkerboard);
// myInit();
// glutMainLoop();
// return 1;
glutInit(&argc, argv); // initialize the toolkit
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB); // set display mode
glutInitWindowSize(640,480); // set window size
//glutInitWindowPosition(100, 150); // set window position on screen glutCreateWindow("null"); //
open the screen window
glutDisplayFunc(myDisplay); // register redraw function myInit();
glutMainLoop();
                }
```

OUTPUT:









LEARNING OUTCOMES:

I learned how to use openGL to construct lines and other shapes in C++.