Name	**:**	**V Vikram**	**Class**	**:**	CSE 'C'
Reg. No.	**:**	**18 5001 194**	**Date**	**:**	23/07/2021
Subject	**:**	**UCS1712---Graphics and Multimedia Lab**			

QUESTION :

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,TRIANGLES and QUADS/POLYGON.

CODE :

1-a.cpp :

```
#include<GL/glut.h>
void myInit_a() {
	glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

	glClearColor(1.0, 1.0, 1.0, 1.0);
	glColor3f(255.0f / 255.0f, 0.0f / 255.0f, 0.0f / 255.0f);
	glPointSize(20);
	glEnable(GL_DEPTH_TEST);

	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}
void myDisplay_a() {
	glClear(GL_COLOR_BUFFER_BIT);
```

```
    /*POINT
    glBegin(GL_POINTS);
    glVertex2d(250,250);*/

    /*LINE
    glBegin(GL_LINES);
    glVertex2f(200, 200);
    glVertex2f(300, 300);*/

    /*LINE_STRIP
    glBegin(GL_LINE_STRIP);
    glVertex2f(200, 200);
    glVertex2f(300, 300);
    glVertex2f(400, 200);*/

    /*LINE_LOOP
    glBegin(GL_LINE_LOOP);
    glVertex2f(100.0f, 100.0f);
    glVertex2f(400.0f, 100.0f);
    glVertex2f(400.0f, 400.0f);
    glVertex2f(100.0f, 400.0f);*/

    /*TRIANGLE
    glBegin(GL_TRIANGLES);
    glVertex2f(100.0f, 100.0f);
    glVertex2f(500.0f, 100.0f);
    glVertex2f(300.0f, 300.0f);*/

    /*QUAD
    glBegin(GL_QUADS);
    glVertex2f(300.0f, 200.0f);
    glVertex2f(400.0f, 300.0f);
    glVertex2f(300.0f, 400.0f);
    glVertex2f(200.0f, 300.0f);*/

    /*QUAD_STRIP
    glBegin(GL_QUAD_STRIP);
    glVertex2f(200.0f, 200.0f);
    glVertex2f(400.0f, 200.0f);
    glVertex2f(400.0f, 400.0f);
    glVertex2f(200.0f, 400.0f);
    glVertex2f(500.0f, 300.0f);*/

    //POLYGON
    glBegin(GL_POLYGON);
    glVertex2f(200.0f, 200.0f);
    glVertex2f(400.0f, 200.0f);
    glVertex2f(400.0f, 300.0f);
    glVertex2f(200.0f, 400.0f);
    glVertex2f(200.0f, 400.0f);

    glEnd();
```

```
	glFlush();
}
int main_a(int argc, char* argv[]) {
	glutInit(&argc, argv);
	glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
	glutInitWindowSize(230,230);
	glutCreateWindow("Ex_1-a");
	glutDisplayFunc(myDisplay_a);
	myInit_a();
	glutMainLoop();
	return 1;
}
```

1-b.cpp :

```
#include<GL/glut.h>
void myInit_b() {
	glClearColor(1.0, 1.0, 1.0, 0.0);
	glPointSize(20);
	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}
int x = 50, y = 50;
bool isRed = true;

void whiteBox(int x, int y)
{
	glBegin(GL_LINE_LOOP);
	glVertex2i(x, y);
	glVertex2i(x, y + 50);
	glVertex2i(x + 50, y + 50);
	glVertex2i(x + 50, y);
	glEnd();
}

void redBox(int x, int y)
{
	glBegin(GL_POLYGON);
	glVertex2i(x, y);
	glVertex2i(x, y + 50);
	glVertex2i(x + 50, y + 50);
	glVertex2i(x + 50, y);
	glEnd();
}
void myDisplay_b(void)
{
```

```
	glClear(GL_COLOR_BUFFER_BIT);
	glColor3f(128.0f / 255.0f, 0.0f / 255.0f, 0.0f / 255.0f);
	glPointSize(1.0);

	for (int i = 0; i < 8; i++)
	{
		if (i % 2 == 0)
		{
			isRed = true;
		}
		else
		{
			isRed = false;
		}

		for (int j = 0; j < 8; j++)
		{
			if (isRed)
			{
				redBox(x, y);
				isRed = false;
			}
			else
			{
				whiteBox(x, y);
				isRed = true;
			}
			x += 50;
		}
		y += 50;
		x = 50;
	}

	redBox(300, 300);
	whiteBox(350, 300);
	glFlush();
}
int main_b(int argc, char* argv[]) {
	glutInit(&argc, argv);
	glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
	glutInitWindowSize(800,600);
	glutCreateWindow("Ex_1-b");
	glutDisplayFunc(myDisplay_b);
	myInit_b();
	glutMainLoop();
	return 1;
}
```

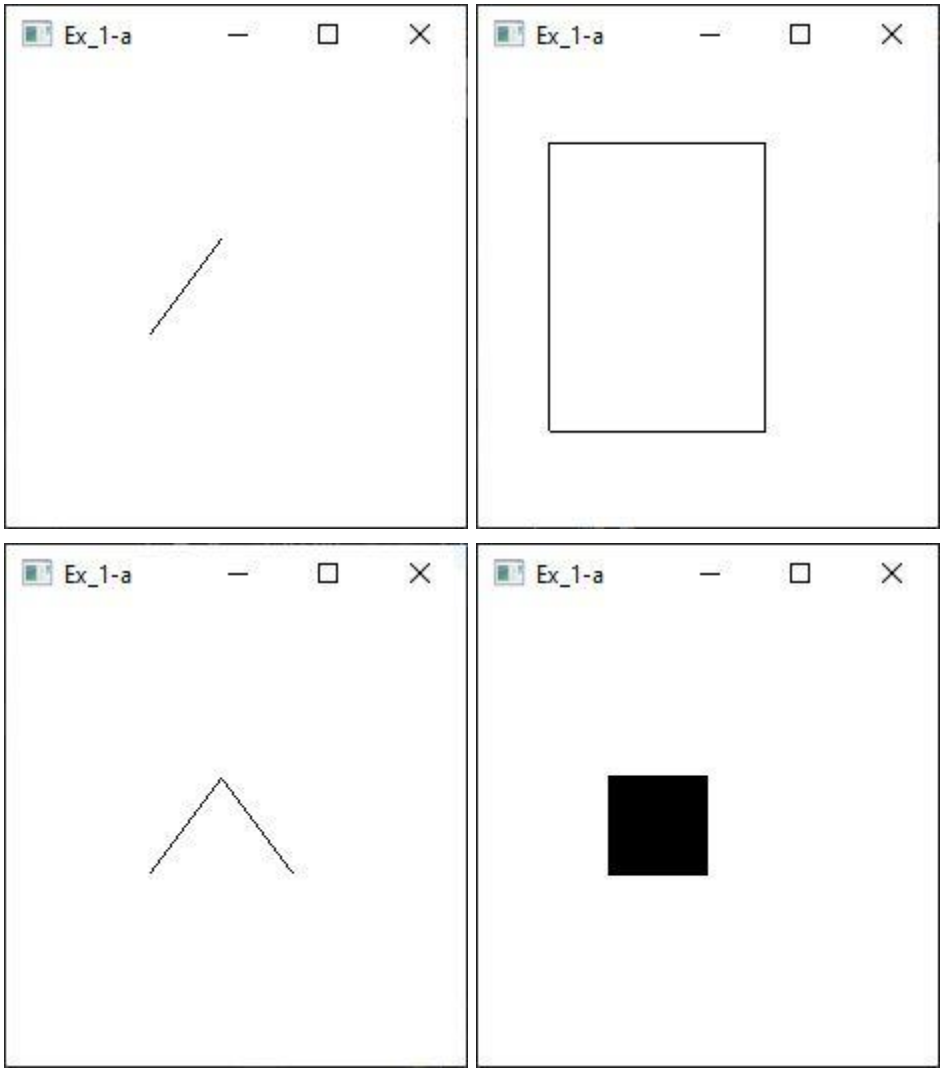
1-c.cpp :

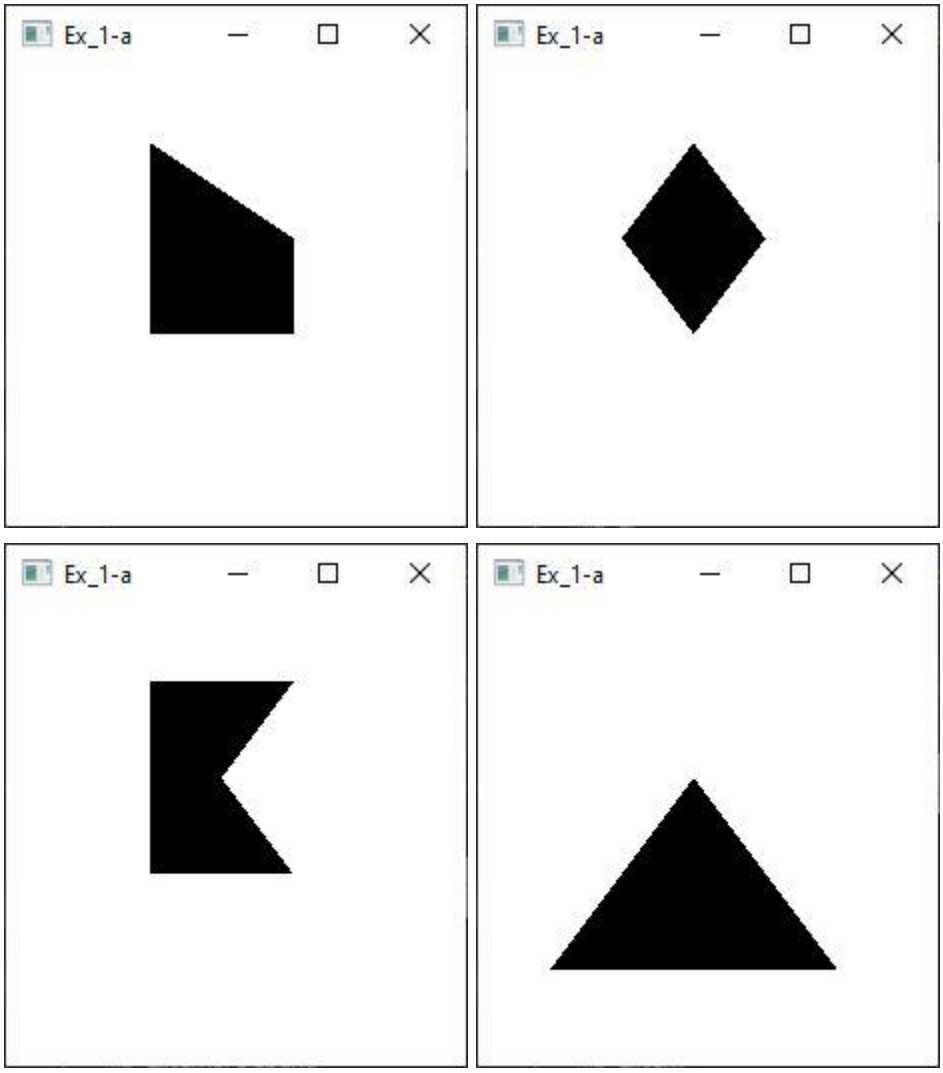
```
#include <GL\glut.h>

void myInit_c(void)
{
    glClearColor(1.0, 1.0, 1.0, 0.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0, 500.0, 0.0, 400.0);

}
void myDisplay_c(void)
{

    glClear(GL_COLOR_BUFFER_BIT); // clears the screen
    glColor3f(0.5f, 0.5f, 0.5f);// sets the drawing colour

    glPointSize(4.0);

    glBegin(GL_POLYGON);

    //bottom
    glVertex2i(20, 20);
    glVertex2i(320, 20);
    glVertex2i(320, 200);
    glVertex2i(20, 200);
    glEnd();


    // Window

    glColor3f(0.0f / 255.0f, 0.0f / 255.0f, 0.0f / 255.0f);
    glPointSize(4.0);
    glRectf(60, 60, 120, 120);

    //Door

    glColor3f(0.0f, 0.0f, 1.0f);
    glBegin(GL_POLYGON);

    glVertex2i(150, 30);
    glVertex2i(210, 30);
    glVertex2i(210, 150);
    glVertex2i(150, 150);


    glEnd();

    //Triangle
    glColor3f(255.0f / 255.0f, 0.0f / 255.0f, 0.0f / 255.0f);
    glBegin(GL_POLYGON);
```

```
    glVertex2i(20, 200);
    glVertex2i(320, 200);
    glVertex2i(170, 310);
    glEnd();

    glFlush(); // sends all output to display;

}
int main(int argc, char** argv)
{

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(640, 480);
    glutInitWindowPosition(10, 10);
    glutCreateWindow("Ex_1-c");

    glutDisplayFunc(myDisplay_c);
    myInit_c();
    glutMainLoop();
    return 0;
}
```

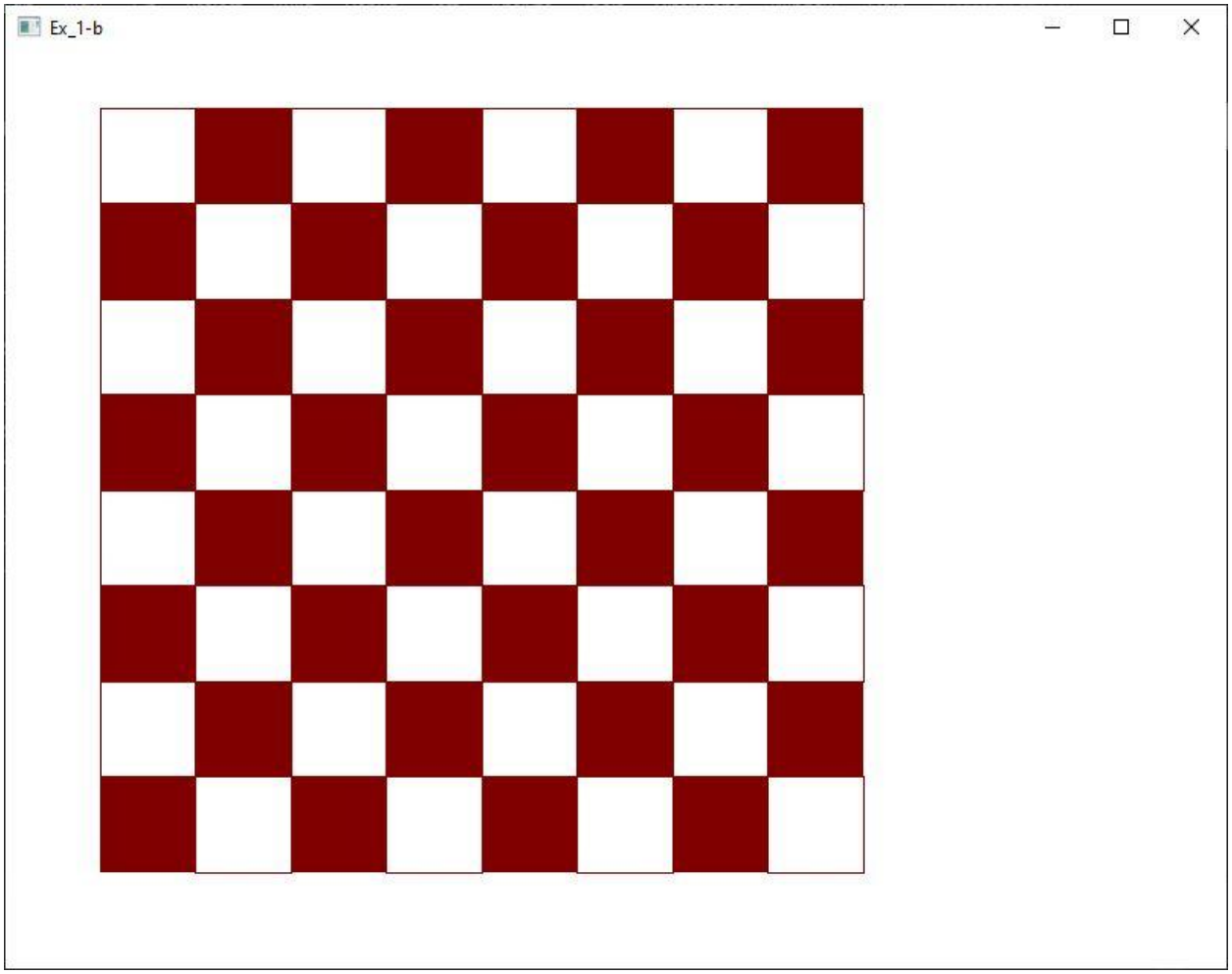
OUTPUT SNAPSHOTS :-

Part-A:

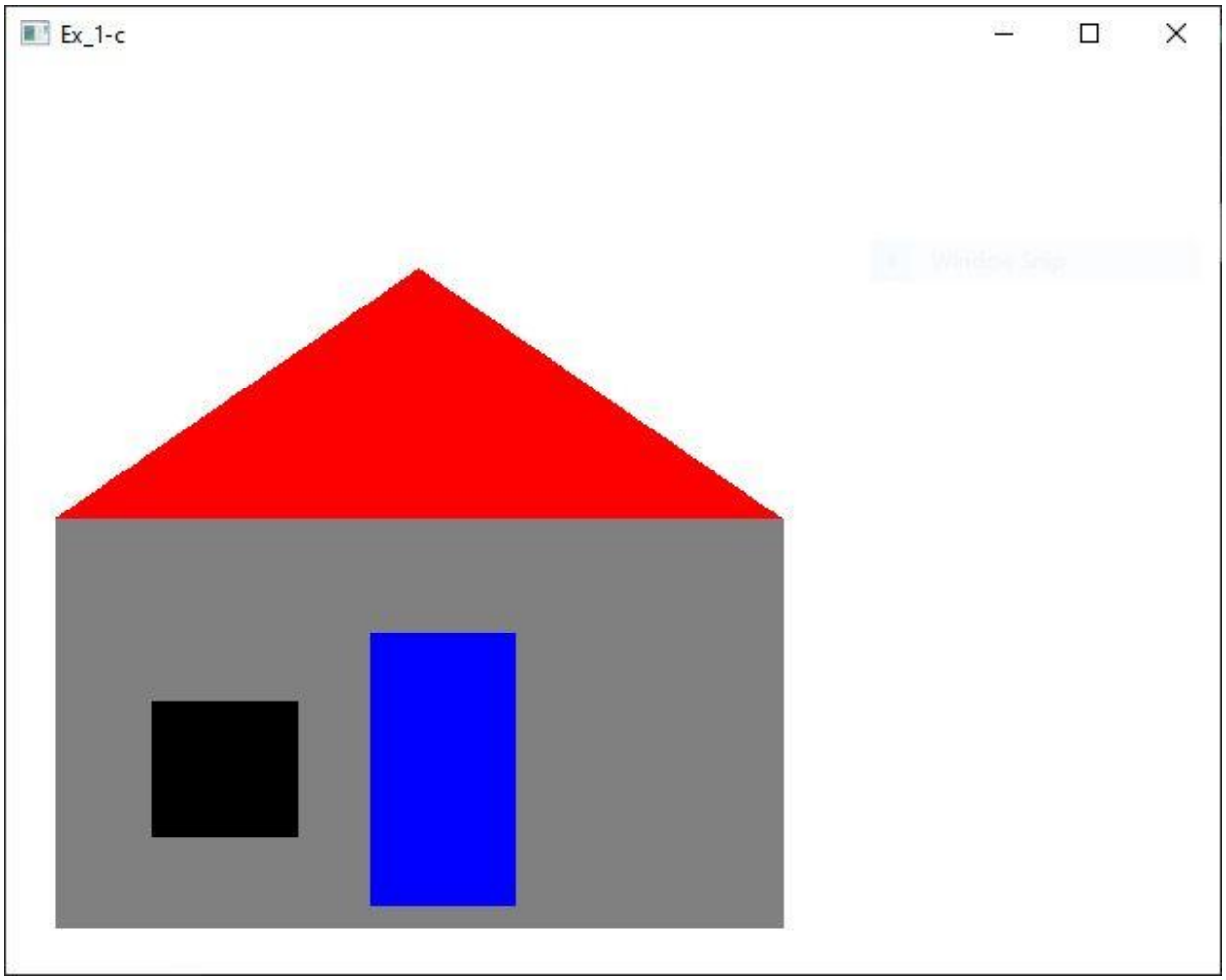



Ex_1-a
Ex_1-a
Ex_1-a
Ex_1-a

Part-B:



Part-C:



CONCLUSION :

Thus the above mentioned basic Output Primitives were studied and implemented in C++ using OpenGL.