NATIONAL UNIVERITY OF MODERN LANGUAGES
Faculty of Engineering & IT

BS(Software Engineering)

Date:_______________

To be filled by student

Subject: **COMPUTER GRAPHICS**

Assignment title: **LAB MANUAL**

Submitted to: **ZAINAB MALIK**

Submitted by

IFTIKHAR AHMAD Roll# **10441**

Program_SemesterBS(SOFTWARE ENGINEERING)-7TH EVENING

To be filled by Teacher:

Total Marks_____________________MarksObtained____________________________________

Remarks____

____

____

Signature___________________

Note: Assignment will not be accepted after due date. Copied and irrelevant assignments will be marked zero. Half marks will be detected for poor formatted assignments.

Tasks:

1) Draw Dot in OpenGL

2) Draw Line in OpenGL

3) Draw Multiple Dots in OpenGL

4) Draw Line Using Functions

5) Draw Circle in OpenGL

6) Draw Triangle in OpenGL

7) Draw Singe Wave in OpenGL

8) Draw EXP in OpenGL

9) Draw Line Using Mouse in OpenGL

10) Draw Line Strip in OpenGL

11) Draw Line Loop in OpenGL

12) Draw Quad in OpenGL

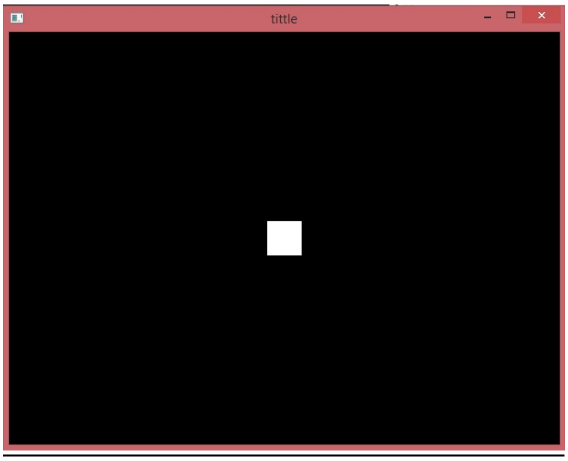
13) Draw Triangle Strip in OpenGL

14) Draw Triangle Fan in OpenGL

15) Draw Quad Strip in OpenGL

16) Draw Polygon in OpenGL

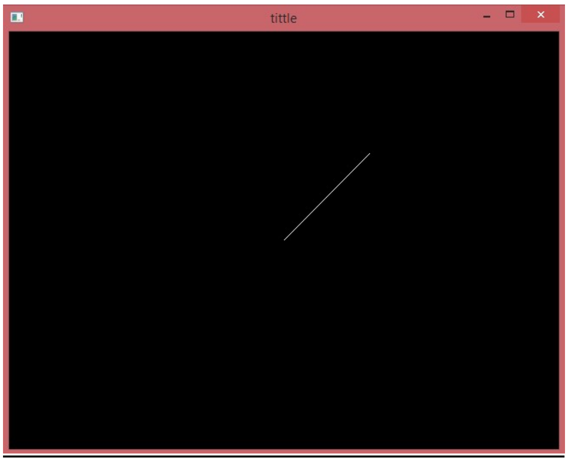
Lab Task 1:



Code:

```
#include<Windows.h>

#include<gl/GL.h>

#include<glut.h>

voidmyInit()

{

glClearColor(0.0, 0.0, 0.0, 0.0);

glColor3f(1.0f, 1.0f, 1.0f);

glPointSize(40.0);

glMatrixMode(GL_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

void display()

{

glClear(GL_COLOR_BUFFER_BIT);

glBegin(GL_POINTS);

glVertex2i(320, 240);

glEnd();

glFlush();

}

int main(intargc, char **argv)

{

glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE);

glutInitWindowSize(640, 480);
glutInitWindowPosition(300, 300);
glutCreateWindow("tittle");

glutDisplayFunc(display);

myInit();

glutMainLoop();

}
```

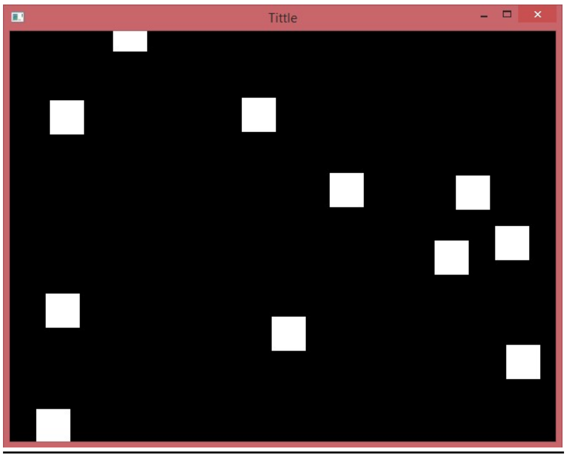
Lab Task 2:



Code:

```
#include<Windows.h>
#include<gl/GL.h>
#include<glut.h>
voidmyInit()
{
        glClearColor(0.0, 0.0, 0.0, 0.0);
        glColor3f(1.0f, 1.0f, 1.0f);
        glPointSize(40.0);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}
void display()
{
        glClear(GL_COLOR_BUFFER_BIT);
        glBegin(GL_LINES);
        glVertex2i(320, 240);
        glVertex2i(420, 340);
        glEnd();
        glFlush();
}
int main(intargc, char **argv)
{
        glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE);
        glutInitWindowSize(640, 480);
glutInitWindowPosition(300, 300);
glutCreateWindow("tittle");
        glutDisplayFunc(display);
        myInit();
        glutMainLoop();
}
```

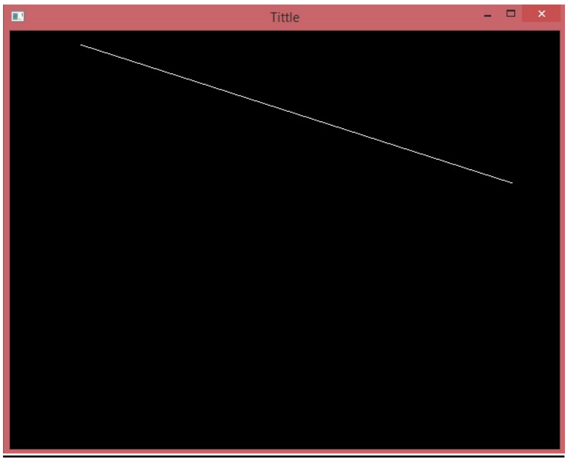
Lab Task 3:



Code:

```
#include<Windows.h>
#include<gl/GL.h>
#include<glut.h>
#include<math.h>

classGlPoint
{
public:
double x;
double y;

public:
        int b;
        int c;
        int d;
        int e;
};

voidmyInit()

{
glClearColor(0.0, 0.0, 0.0, 0.0);
glColor3f(1.0f, 1.0f, 1.0f);
glPointSize(40.0);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}
voiddisplaydot(GlPointa) {
        glBegin(GL_POINTS);
        glVertex2d(a.x, a.y);
        glEnd();
}

GlPoint p1;
void display()
{
        glClear(GL_COLOR_BUFFER_BIT);
        for (inti = 0; i<= 10;i++)
        {
        p1.x = rand()%640;
p1.y = rand()%480;
        displaydot(p1);
        }
        glFlush();
}
int main(intargc, char **argv)
{
glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_SINGLE);
        glutInitWindowSize(640, 480);
        glutInitWindowPosition(300, 300);

        glutCreateWindow("Tittle");
        glutDisplayFunc(display);

        myInit();
        glutMainLoop();

}
```

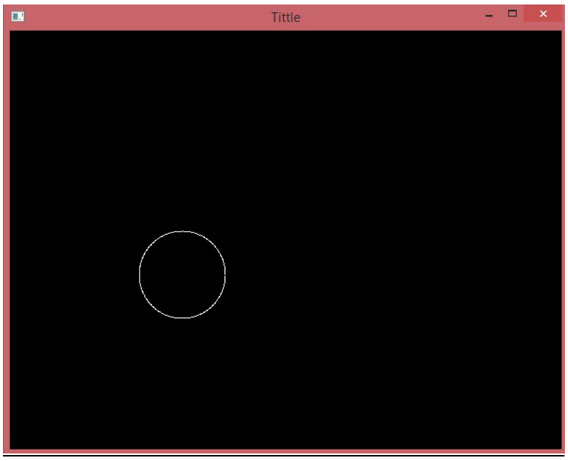
Lab Task 4:



Code:

```
#include<Windows.h>
#include<gl/GL.h>
#include<glut.h>
#include<math.h>
classGlPoint
{
public:                         double x;
        double y;

public:
        int b;
        int c;
        int d;
        int e;
};

voidmyInit()

{glClearColor(0.0, 0.0, 0.0, 0.0);
glColor3f(1.0f, 1.0f, 1.0f);
glPointSize(40.0);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}
voiddisplaydot(GlPointa) {
        glBegin(GL_POINTS);
        glVertex2d(a.x, a.y);
        glEnd();
}
voiddrawline(GlPointa)
{
        glBegin(GL_LINES);
        glVertex2i(a.b, a.c);
        glVertex2i(a.d, a.e);
        glEnd();
}

GlPoint p1;

void display()
{
        glClear(GL_COLOR_BUFFER_BIT);
        p1.b = rand() % 640;
        p1.c= rand() % 480;
        p1.d = rand() % 640;
        p1.e = rand() % 480;
        drawline(p1);

        glFlush();

}

int main(intargc, char **argv)
{

        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_SINGLE);
        glutInitWindowSize(640, 480);
        glutInitWindowPosition(300, 300);

        glutCreateWindow("Tittle");
        glutDisplayFunc(display);

        myInit();
        glutMainLoop();

}
```

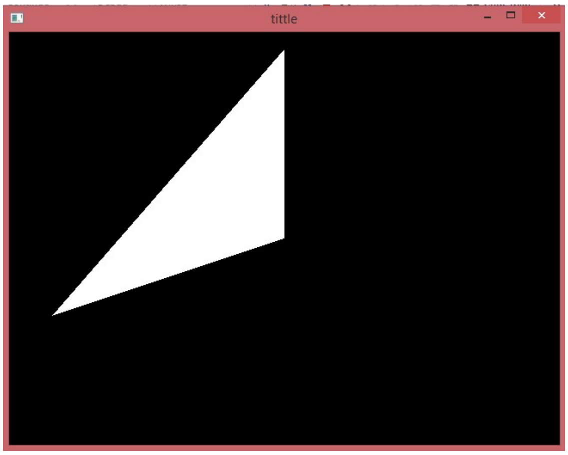
Lab Task 5:



Code:

```
#include<Windows.h>
#include<gl/GL.h>
#include<glut.h>
#include<math.h>
classGlPoint
{
public:                                 double x;
          double y;
public:
          int b;
          int c;
          int d;
          int e;
};
voidmyInit()
{glClearColor(0.0, 0.0, 0.0, 0.0);
glColor3f(1.0f, 1.0f, 1.0f);
//glPointSize(40.0);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}
voiddisplaydot(GlPointa) {
          glBegin(GL_POINTS);
          glVertex2d(a.x, a.y);
          glEnd();
}
voiddrawline(GlPointa)
{
          glBegin(GL_LINES);
          glVertex2i(a.b, a.c);
          glVertex2i(a.d, a.e);
          glEnd();
}

GlPoint p1;

void display()
{
          glClear(GL_COLOR_BUFFER_BIT);
          for (double angle = 0.0174; angle <= 
0.0174 * 360; angle += 0.0174)
          {
                    p1.x = 50 * cos(angle) + 200;
                    p1.y = 50 * sin(angle) + 200;
                    displaydot(p1);
          }         drawline(p1);

          glFlush();

}

int main(intargc, char **argv)
{

          glutInit(&argc, argv);
          glutInitDisplayMode(GLUT_SINGLE);
          glutInitWindowSize(640, 480);
          glutInitWindowPosition(300, 300);
          glutCreateWindow("Tittle");
          glutDisplayFunc(display);
          myInit();
          glutMainLoop();

}
```

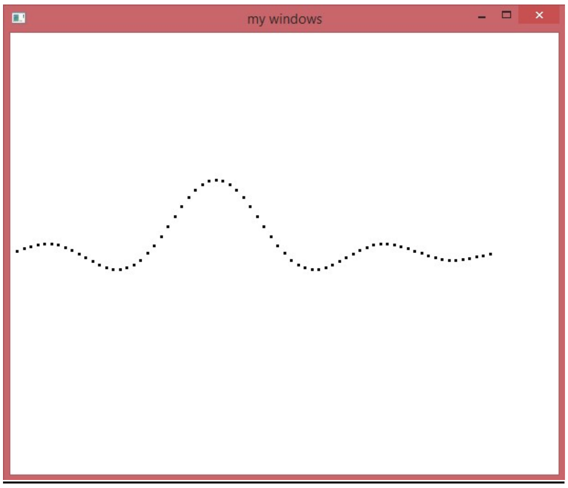
Lab Task 6:



Code:

```
#include<Windows.h>

#include<gl/GL.h>

#include<glut.h>

voidmyInit()

{

        glClearColor(0.0, 0.0, 0.0, 0.0);
        glColor3f(1.0f, 1.0f, 1.0f);
        glPointSize(40.0);

        glMatrixMode(GL_PROJECTION);

        glLoadIdentity();

        gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

void display()

{

        glClear(GL_COLOR_BUFFER_BIT);
        glBegin(GL_TRIANGLES);
        glVertex2i(50, 150);
        glVertex2i(320, 240);
        glVertex2i(320, 460);
        glEnd();

        glFlush();}



int main(intargc, char **argv)

{

        glutInit(&argc, argv);

        glutInitDisplayMode(GLUT_SINGLE);

        glutInitWindowSize(640, 480);
        glutInitWindowPosition(300, 300);
        glutCreateWindow("tittle");

        glutDisplayFunc(display);

        myInit();
        glutMainLoop();

}
```

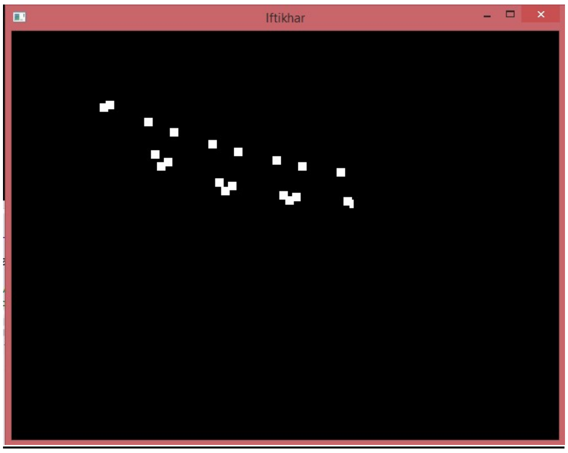
Lab Task 7:



Code:

```
#include<windows.h>
#include<gl/GL.h>
#include<glut.h>
#include<math.h>
float a, b, c, d;
voidmyInit()
{
        glClearColor(1.0, 1.0, 1.0, 0.0);
        glColor3f(0.0f, 0.0f, 0.0f);
        glPointSize(3.0);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        gluOrtho2D(0.0, 640.0, 0.0, 480.0);
        a = 640 / 8;
        b = 480 / 2;
        c = 640 / 8;
        d = 480 / 2;

}


void display()
{
        glClear(GL_COLOR_BUFFER_BIT);
        glBegin(GL_POINTS);
        for (float x = -4.0; x < 4.0; x += 0.1)
        {
                float y = sin(3.1415*x) /
(3.1415*x);
                glVertex2f(x*a + b, y*c + d);
        }
        glEnd();
        glFlush();

}
int main(intargc, char **argv)
{
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_SINGLE);
        glutInitWindowSize(600, 800);
        glutInitWindowPosition(0, 0);
        glutCreateWindow("my windows");
        glutDisplayFunc(display);
        myInit();
        glutMainLoop();
}
```

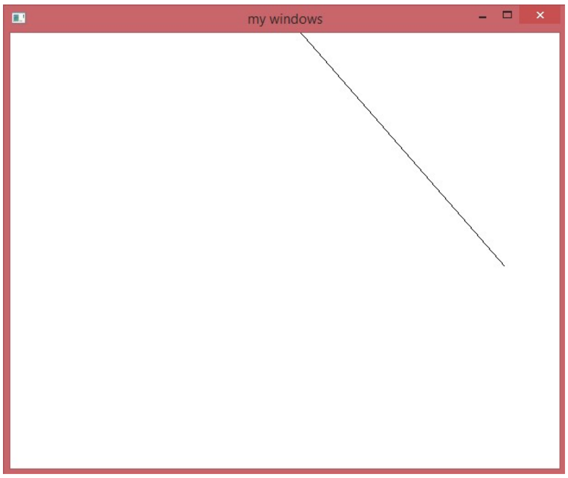
Lab Task 8:



Code:

```
#include<Windows.h>
#include<gl/GL.h>
#include<glut.h>
#include<math.h>
voidmyInit()
{

	glClearColor(0.0, 0.0, 0.0, 0.0);
	glColor3f(1.0f, 1.0f, 1.0f);
	glPointSize(10.0);
}
voidsetwindow(floatwleft, floatwright, floatwbotton, floatwtop)
{

	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	gluOrtho2D(wleft, wright, wbotton, wtop); }
voidsetviewport(floatvleft, floatvright, floatvbotton, floatvtop)
{
	glViewport(vleft, vbotton, vright - vleft, vtop - vbotton);
}
void display()
{
	glClear(GL_COLOR_BUFFER_BIT);
	setwindow(0.0, 4.0, -1.0, 1.0);
	setviewport(100, 400, 150, 400);
	glBegin(GL_POINTS);
	for (float x = -4.0; x < 4.0; x += 0.1)
	{
		float y = exp(-fabs(x)*cos(2 * 3.1415*x));
		glVertex2f(x, y);
	}
	glEnd();
	glFlush();
}
int main(intargc, char **argv) {
	glutInit(&argc, argv);
	glutInitDisplayMode(GLUT_SINGLE);
	glutInitWindowSize(640, 480);
	glutInitWindowPosition(300, 300);
	glutCreateWindow("Iftikhar");
	glutDisplayFunc(display);
	myInit();
	glutMainLoop();
}
```

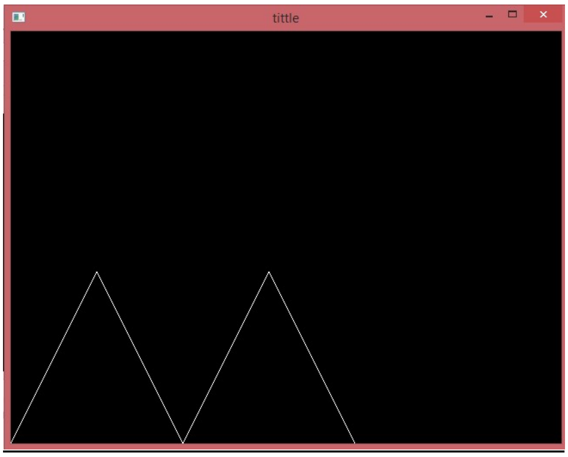
Lab Task 9:



Code:

```
#include<Windows.h>
#include<gl/GL.h>
#include<glut.h>
classGLPoint
{
public:
	float x;
	float y;
};
voidmyInit()
{glClearColor(1.0, 1.0, 1.0, 0.0);
glColor3f(0.0f, 0.0f, 0.0f);
glPointSize(3.0);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}
GLPoint start, finish;
void display()
{
glClear(GL_COLOR_BUFFER_BIT);
glBegin(GL_LINES);
glVertex2f(start.x, start.y);
glVertex2f(finish.x, finish.y);
glEnd();
glFlush();
}
void mouse(intbutton, intstate, intmousex,
intmousey)
{
if (button == GLUT_LEFT_BUTTON&&state
== GLUT_DOWN)
{
start.x = mousex;
start.y = 800 - mousey;
glFlush();
}
elseif (button ==
GLUT_LEFT_BUTTON&&state == GLUT_UP)
{
finish.x = mousex;
finish.y = 800 - mousey;
glutPostRedisplay();
}
elseif (button ==
GLUT_RIGHT_BUTTON&&state ==
GLUT_DOWN)
{exit(-1);}
}
int main(intargc, char **argv)
{
glutInit(&argc,
argv);glutInitDisplayMode(GLUT_SINGLE);gl
utInitWindowSize(600, 800);
glutInitWindowPosition(0, 0);
glutCreateWindow("my windows");
glutDisplayFunc(display);glutMouseFunc(m
ouse);
myInit();
glutMainLoop();

}
```

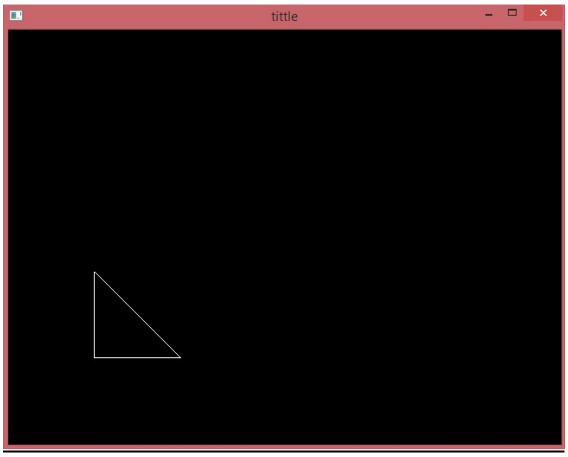
Lab Task 10:



Code:

```
#include<Windows.h>
#include<gl/GL.h>
#include<glut.h>
voidmyInit()
{
	glClearColor(0.0, 0.0, 0.0, 0.0);
	glColor3f(1.0f, 1.0f, 1.0f);
	glPointSize(40.0);
	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}
void display()
{
	glClear(GL_COLOR_BUFFER_BIT);
	glBegin(GL_LINE_STRIP);

	glVertex2i(0.0, 0.00);
	glVertex2i(100, 200);
	glVertex2i(200, 000);
	glVertex2i(300, 200);
	glVertex2i(400, 000);

	glEnd();
	glFlush();
}
int main(intargc, char **argv)
{
	glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE);
	glutInitWindowSize(640, 480);
glutInitWindowPosition(300, 300);
glutCreateWindow("tittle");
	glutDisplayFunc(display);
	myInit();
	glutMainLoop();
}
```

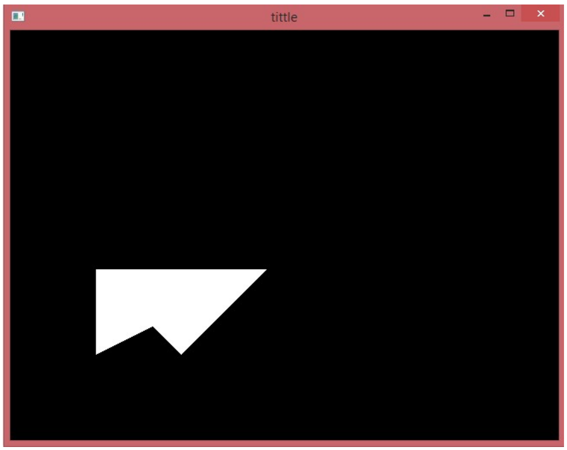
Lab Task 11:



Code:

```
#include<Windows.h>
#include<gl/GL.h>
#include<glut.h>
voidmyInit()
{
        glClearColor(0.0, 0.0, 0.0, 0.0);
        glColor3f(1.0f, 1.0f, 1.0f);
        glPointSize(40.0);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}
void display()
{
        glClear(GL_COLOR_BUFFER_BIT);
        glBegin(GL_LINE_LOOP);

        glVertex2i(100, 100);
        glVertex2i(100, 200);
        glVertex2i(200, 100);
        //glVertex2i(300, 200);
        //glVertex2i(400, 000);

        glEnd();
        glFlush();
}
int main(intargc, char **argv)
{
        glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE);
        glutInitWindowSize(640, 480);
glutInitWindowPosition(300, 300);
glutCreateWindow("tittle");
        glutDisplayFunc(display);
        myInit();
        glutMainLoop();
}
```

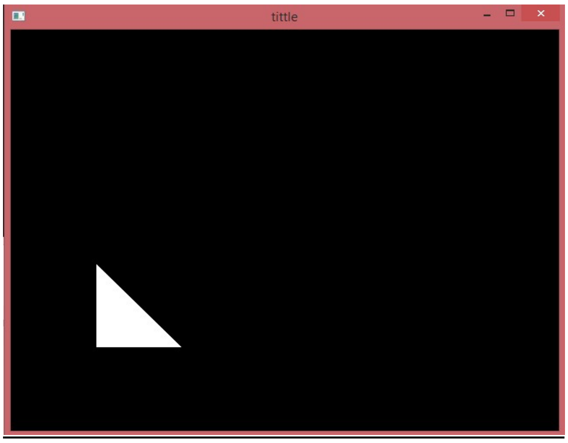
Lab Task 12:



Code:

```
#include<Windows.h>
#include<gl/GL.h>
#include<glut.h>
voidmyInit()
{
        glClearColor(0.0, 0.0, 0.0, 0.0);
        glColor3f(1.0f, 1.0f, 1.0f);
        glPointSize(40.0);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}
void display()
{
        glClear(GL_COLOR_BUFFER_BIT);
        glBegin(GL_QUADS);

        glVertex2i(100, 100);
        glVertex2i(100, 200);
        glVertex2i(200, 100);
        glVertex2i(300, 200);

        glEnd();
        glFlush();
}
int main(intargc, char **argv)
{
        glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE);
        glutInitWindowSize(640, 480);
glutInitWindowPosition(300, 300);
glutCreateWindow("tittle");
        glutDisplayFunc(display);
        myInit();
        glutMainLoop();
}
```

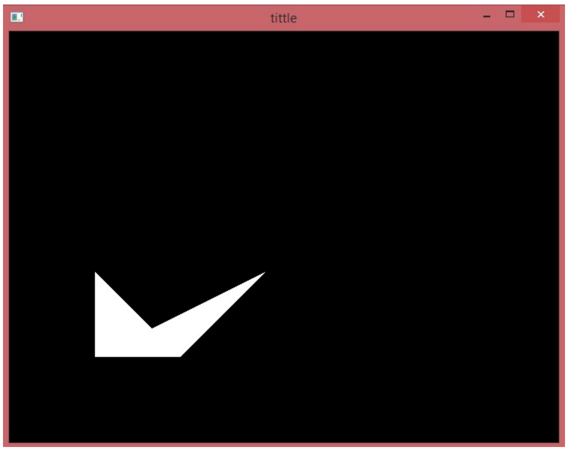
Lab Task 13:



Code:

```
#include<Windows.h>
#include<gl/GL.h>
#include<glut.h>
voidmyInit()
{
        glClearColor(0.0, 0.0, 0.0, 0.0);
        glColor3f(1.0f, 1.0f, 1.0f);
        glPointSize(40.0);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}
void display()
{
        glClear(GL_COLOR_BUFFER_BIT);
        glBegin(GL_TRIANGLE_STRIP);

        glVertex2i(100, 100);
        glVertex2i(100, 200);
        glVertex2i(200, 100);

        glEnd();
        glFlush();
}
int main(intargc, char **argv)
{
        glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE);
        glutInitWindowSize(640, 480);
glutInitWindowPosition(300, 300);
glutCreateWindow("tittle");
        glutDisplayFunc(display);
        myInit();
        glutMainLoop();
}
```

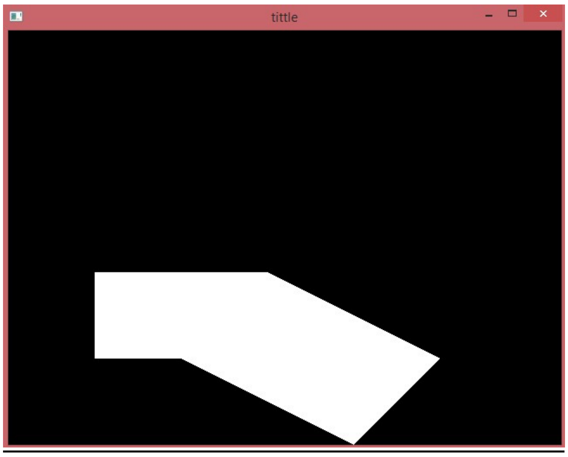
Lab Task 14:



Code:

```
#include<Windows.h>
#include<gl/GL.h>
#include<glut.h>
voidmyInit()
{
	glClearColor(0.0, 0.0, 0.0, 0.0);
	glColor3f(1.0f, 1.0f, 1.0f);
	glPointSize(40.0);
	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}
void display()
{
	glClear(GL_COLOR_BUFFER_BIT);
	glBegin(GL_TRIANGLE_FAN);

	glVertex2i(100, 100);
	glVertex2i(100, 200);
	glVertex2i(200, 100);
	glVertex2i(300, 200);

	glEnd();
	glFlush();
}
int main(intargc, char **argv)
{
	glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE);
	glutInitWindowSize(640, 480);
glutInitWindowPosition(300, 300);
glutCreateWindow("tittle");
	glutDisplayFunc(display);
	myInit();
	glutMainLoop();
}
```

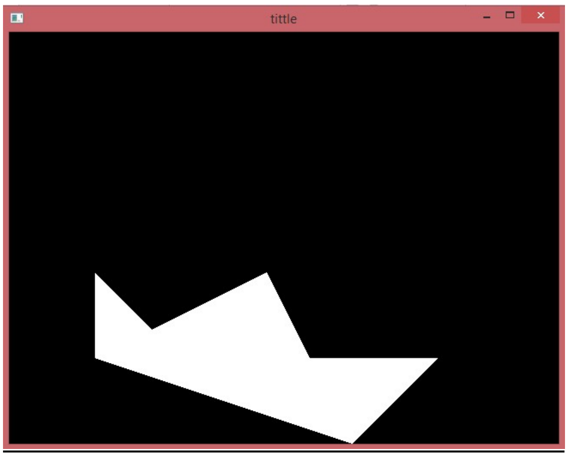
Lab Task 15:



Code:

```
#include<Windows.h>
#include<gl/GL.h>
#include<glut.h>
voidmyInit()
{
	glClearColor(0.0, 0.0, 0.0, 0.0);
	glColor3f(1.0f, 1.0f, 1.0f);
	glPointSize(40.0);
	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}
void display()
{
	glClear(GL_COLOR_BUFFER_BIT);
	glBegin(GL_QUAD_STRIP);
	glVertex2i(100, 100);
	glVertex2i(100, 200);
	glVertex2i(200, 100);
	glVertex2i(300, 200);
	glVertex2i(400, 000);
	glVertex2i(500, 100);

	glEnd();
	glFlush();
}
int main(intargc, char **argv)
{
	glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE);
	glutInitWindowSize(640, 480);
glutInitWindowPosition(300, 300);
glutCreateWindow("tittle");
	glutDisplayFunc(display);
	myInit();
	glutMainLoop();
}
```

Lab Task 16:



Code:

```
#include<Windows.h>
#include<gl/GL.h>
#include<glut.h>
voidmyInit()
{
	glClearColor(0.0, 0.0, 0.0, 0.0);
	glColor3f(1.0f, 1.0f, 1.0f);
	glPointSize(40.0);
	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}
void display()
{
	glClear(GL_COLOR_BUFFER_BIT);
	glBegin(GL_POLYGON);

	glVertex2i(100, 100);
	glVertex2i(100, 200);
	glVertex2i(200, 100);
	glVertex2i(300, 200);
	glVertex2i(400, 000);
	glVertex2i(500, 100);

	glEnd();
	glFlush();
}
int main(intargc, char **argv)
{
	glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE);
	glutInitWindowSize(640, 480);
glutInitWindowPosition(300, 300);
glutCreateWindow("tittle");
	glutDisplayFunc(display);
	myInit();
	glutMainLoop();
}
```