

## الجلسة الثانية: رسم الأشكال الأساسية

### الهدف من الجلسة

- التعرف ببيئة الرسم.
- رسم بعض الأشكال الأساسية.

## بيئة الرسم الأساسية:

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

```
void myDisplay(void)
```

```
{
```

```
//Draw Instructions
```

```
}
```

```
void main (int argc, char **argv)
```

```
{
```

```
glutInit (&argc, argv); // to initialize the toolkit;
```

```
glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB); //sets the display mode
```

```
glutInitWindowSize (640, 480); // sets the window size
```

```
glutInitWindowPosition (10, 10); // sets the starting position for the window
```

```
glutCreateWindow ("My first OpenGL program!"); //creates the window and sets the title
```

```
glutDisplayFunc (myDisplay);
```

```
glutMainLoop(); // go into a loop until event occurs
```

```
}
```

استخدام **buffer** واحد للعرض

يستخدم مع نموذج الألوان  
**RED GREEN BLUE**

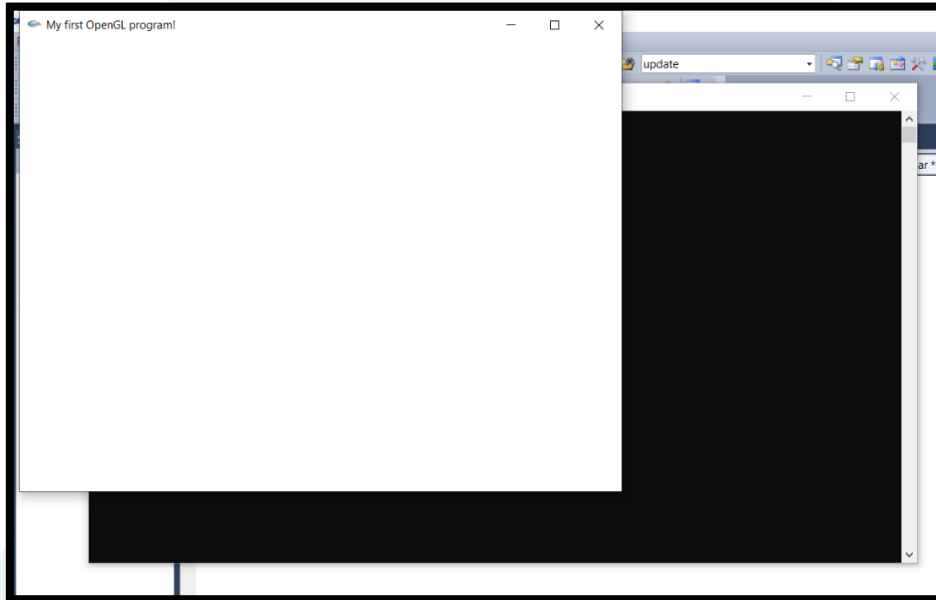
عرض  
شاشة  
الرسم

طول شاشة  
الرسم

موقع شاشة  
الرسم بالنسبة  
لشاشة  
الحاسب

استدعاء تابع الرسم

## نتيجة تنفيذ الكود السابق:



## تغيير لون الخلفية:

```
#include<GL/glut>
void myDisplay(void)
{
glClearColor(0,0,0,0);
```

جعل الخلفية سوداء

alpha

```
glClear(GL_COLOR_BUFFER_BIT);
glFlush();
}
```

تصفير BIT الألوان

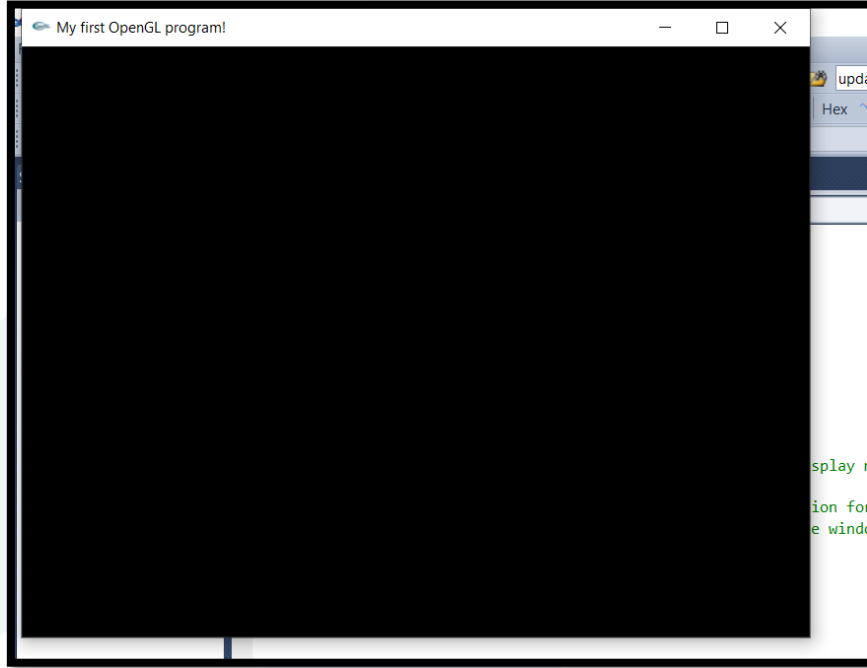
لتنفيذ الكود السابق

```
void main (int argc, char **argv)
```

```
{
glutInit (&argc, argv); // to initialize the toolkit;
glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB); //sets the display mode
glutInitWindowSize (640, 480); // sets the window size
glutInitWindowPosition (10, 10); // sets the starting position for the window
glutCreateWindow ("My first OpenGL program!"); //creates the window and sets the title
```

```
glutDisplayFunc (myDisplay);  
glutMainLoop(); // go into a loop until event occurs  
}
```

نتيجة تنفيذ الكود السابق:



رموز الألوان المختلفة:

1. أحمر: 1,0,0
2. أخضر: 0,1,0
3. أزرق: 0,0,1
4. أصفر: 1,1,0
5. أزرق فاتح: 0,1,1
6. أرجواني: 1,0,1
7. برتقالي: 1,0.5,0
8. وردي: 1,0,0.5
9. رمادي: 0.5,0.5,0.5
10. أبيض: 1,1,1
11. أسود: 0.0,0.0,0.0

## رسم الأشكال في فضاء ثنائي البعد:

يتم الرسم دائماً بين التابع glBegin() وتابع glEnd()، وتحدد إحداثيات الرسم بتابع glVertex ولون الشكل بتابع glColor بالشكل التالي:

**glBegin(shape\_to\_draw);**

glColor3f(red, green, blue);

glVertex2d(x,y);

**glEnd();**

ملاحظة: يوضع البلوك السابق في تابع الرسم.

### رسم مستقيم:

- نستخدم نوع الشكل GL\_LINES.
- نحتاج إلى إحداثيات البداية والنهاية.

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

```
void myDisplay(void)
```

```
{
```

```
glClearColor(0.5,0.5,0.5,0);
```

```
glClear(GL_COLOR_BUFFER_BIT);
```

```
glBegin(GL_LINES);
```

```
glColor3f(1,0,0);
```

```
glVertex2d(-1,0);
```

```
glVertex2d(1,1);
```

لون الشكل

يقبل قيم float

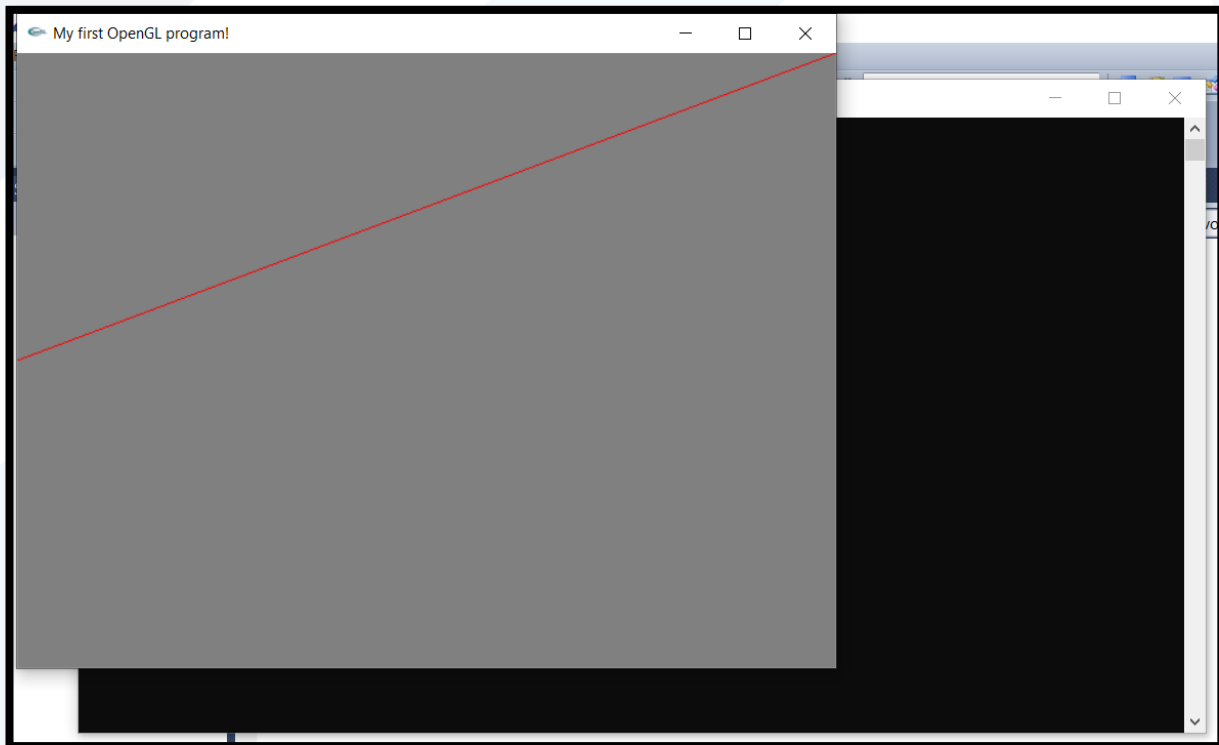
نقطة البداية

5

نقطة النهاية

```
glEnd();  
glFlush();  
}  
void main (int argc, char **argv)  
{  
    glutInit (&argc, argv); // to initialize the toolkit;  
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB); //sets the display mode  
    glutInitWindowSize (640, 480); // sets the window size  
    glutInitWindowPosition (10, 10); // sets the starting position for the window  
    glutCreateWindow ("My first OpenGL program!"); //creates the window and sets the title  
    glutDisplayFunc (myDisplay);  
    glutMainLoop(); // go into a loop until event occurs  
}
```

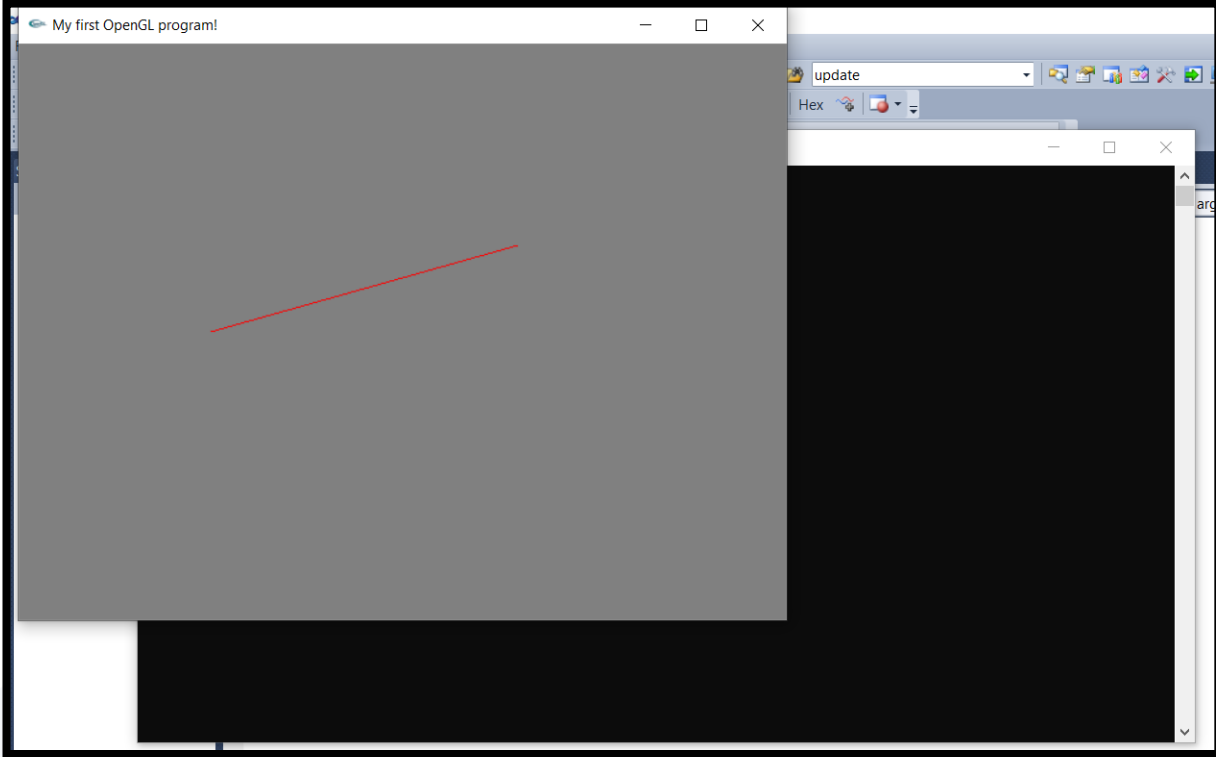
نتائج تنفيذ الكود التالي:



لجعل قيم نقطة البداية والنهاية أكبر نستخدم تابع (`gluOrtho2D(x_start,x_end,y_start,y_end)`)

```
#include<GL/glut.h>
void myDisplay(void)
{
glClearColor(0.5,0.5,0.5,0);
glClear(GL_COLOR_BUFFER_BIT);
glBegin(GL_LINES);
glColor3f(1,0,0);
glVertex2d(-5,0);
glVertex2d(3,3);
glEnd();
glFlush();
}
void main (int argc, char **argv)
{
glutInit (&argc, argv); // to initialize the toolkit;
glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB); //sets the display mode
glutInitWindowSize (640, 480); // sets the window size
glutInitWindowPosition (10, 10); // sets the starting position for the window
glutCreateWindow ("My first OpenGL program!"); //creates the window and sets the title
gluOrtho2D(-10,10,-10,10);
glutDisplayFunc (myDisplay);
glutMainLoop(); // go into a loop until event occurs
}
```

## خرج الكود السابق:

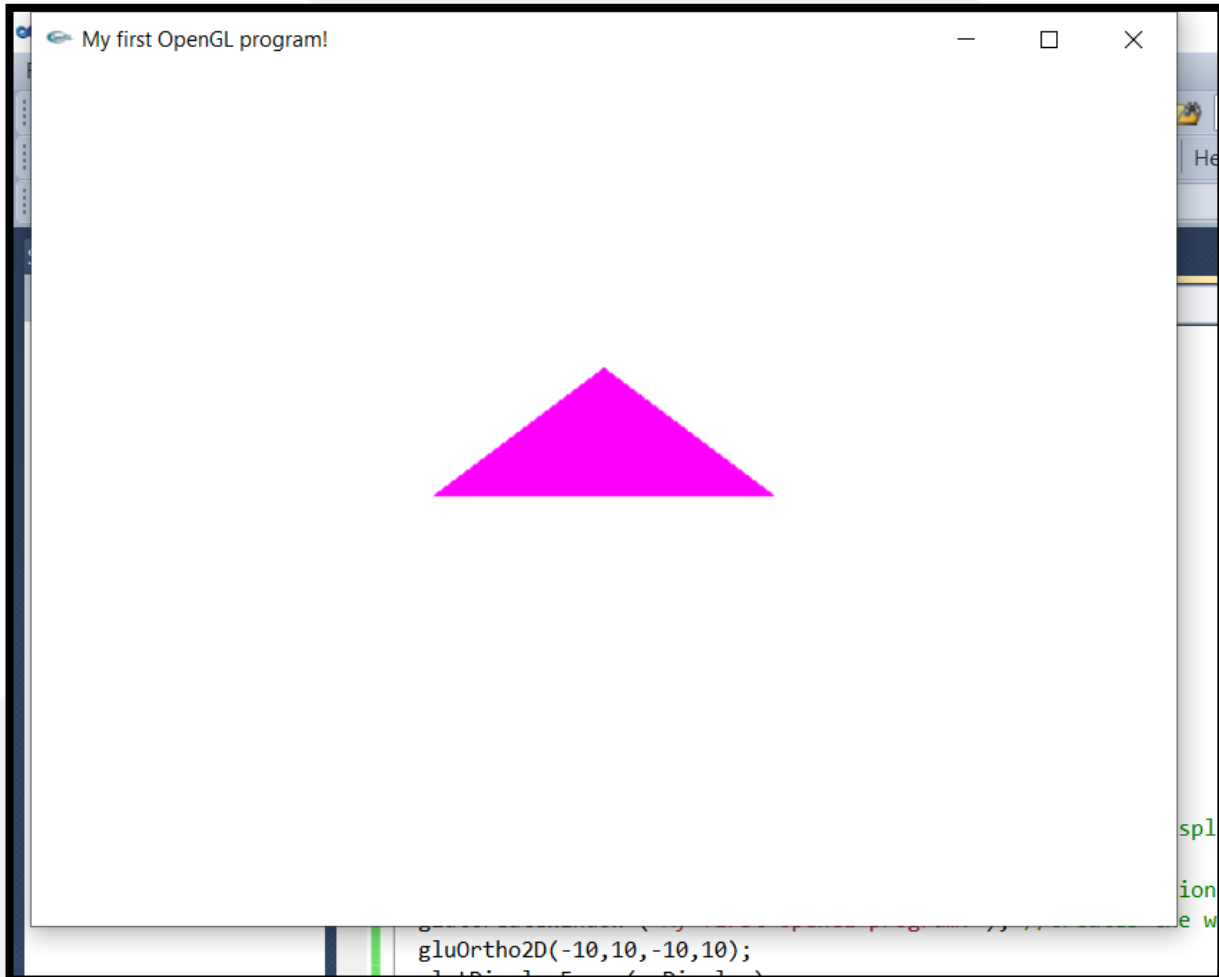


### رسم المثلث:

- نستخدم نوع الشكل `GL_TRIANGLES`.
- نحتاج إلى إحداثيات البداية والنهاية.

```
void myDisplay(void)
{
    glClearColor(1,1,1,0);
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_TRIANGLES);
    glColor3f(1,0,1);
    glVertex2d(-3,0);
    glVertex2d(0,3);
    glVertex2d(3,0);
    glEnd();
    glFlush();
}
```

خرج الكود السابق:



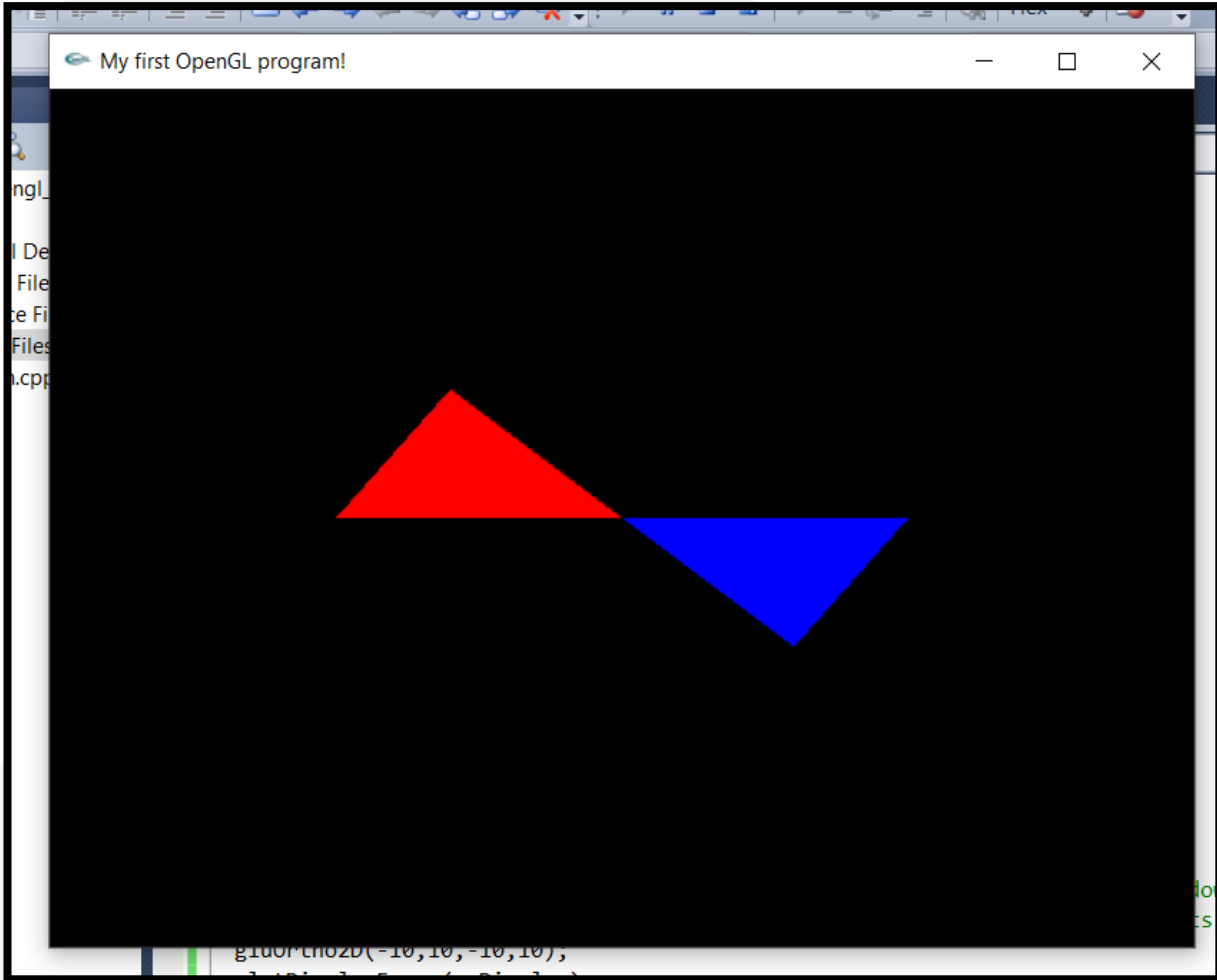
الأشكال الأخرى:

الوصف	NAME
لرسم خط .. وهنا يلزمنا نقطتين	GL_LINES
رسم خطوط متصلة	GL_LINE_STRIP
نفس الأمر السابق لكن يتم وصل آخر نقطة بأول نقطة ليتم رسم شكل مغلق	GL_LINE_LOOP
لرسم مثلث .. وتحتاج إلى ثلاث نقاط أو مضاعفات الثلاثة	GL_TRIANGLES
لرسم مثلثات متصلة	GL_TRIANGLE_STRIP
لرسم مثلثات تتصل بنقطة المنتصف (غالبا تستخدم لرسم الأشكال الدائرية)	GL_TRIANGLE_FAN
لرسم أشكال مربعة أو مستطيلة (يعني أشكال ذات أربع رؤوس) وتحتاج إلى أربع نقاط لرسم شكل رباعي واحد .. وثمان نقاط لرسم شكلين رباعيين	GL_QUADS
لرسم أشكال رباعية متصلة	GL_QUAD_STRIP
لرسم مضلع .	GL_POLYGON

## تمارين:

تمرين 1:

1-1 ارسم الشكل التالي:



2-1 اجعل إحاثيات المثلث الأحمر كالتالي:

```
glVertex2d(15,0);
```

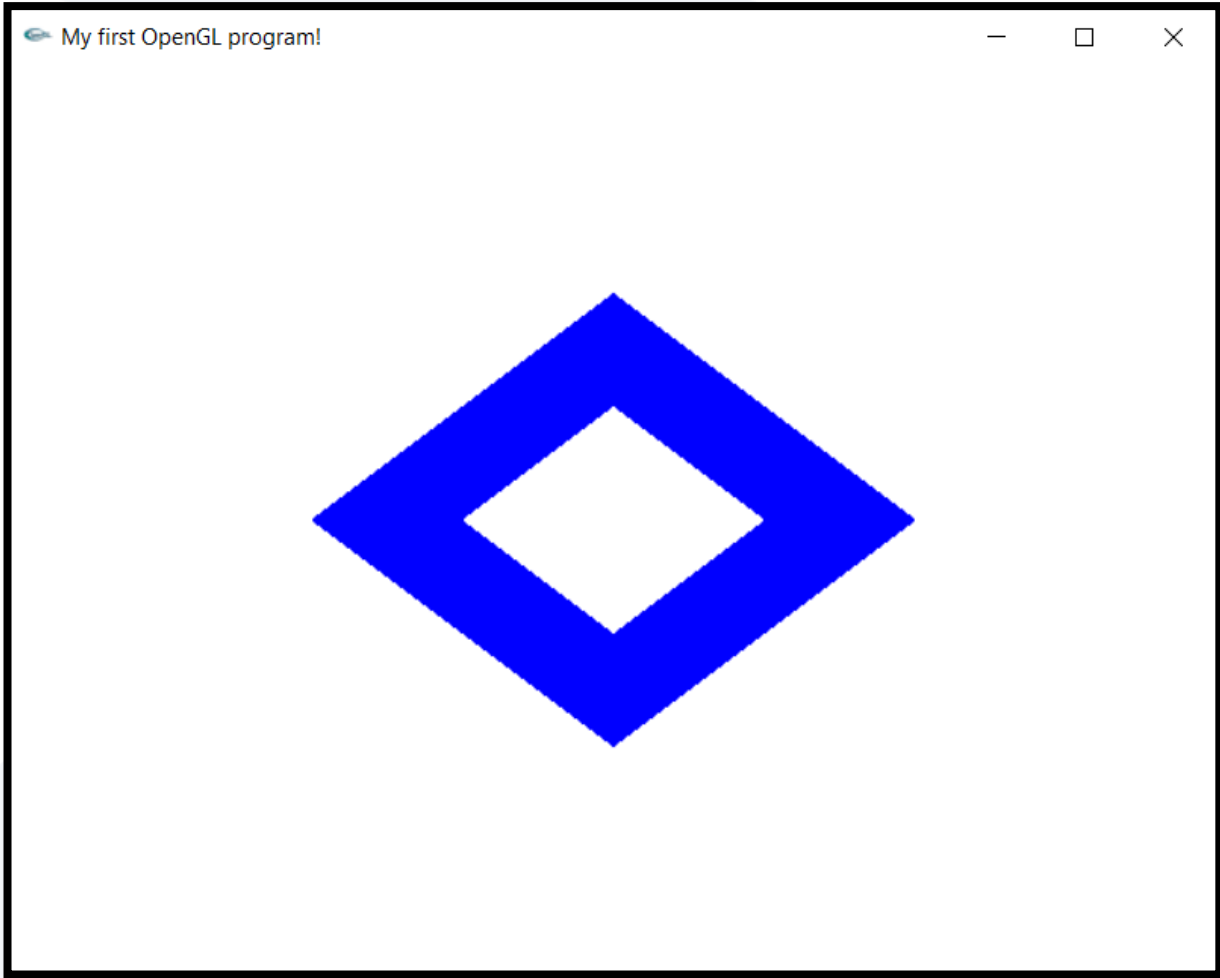
```
glVertex2d(13,-11);
```

```
glVertex2d(0,0);
```

3-1 أجري ما يلزم لجعل الشكل ظاهراً بشكل كامل على شاشة الرسم.

تمرين 2:

1-2 ارسم الشكل التالي:



الحل:

[https://drive.google.com/drive/folders/1at96q\\_Lx-xUFb9Zr1mPpPpEROZBQCx4v?usp=sharing](https://drive.google.com/drive/folders/1at96q_Lx-xUFb9Zr1mPpPpEROZBQCx4v?usp=sharing)