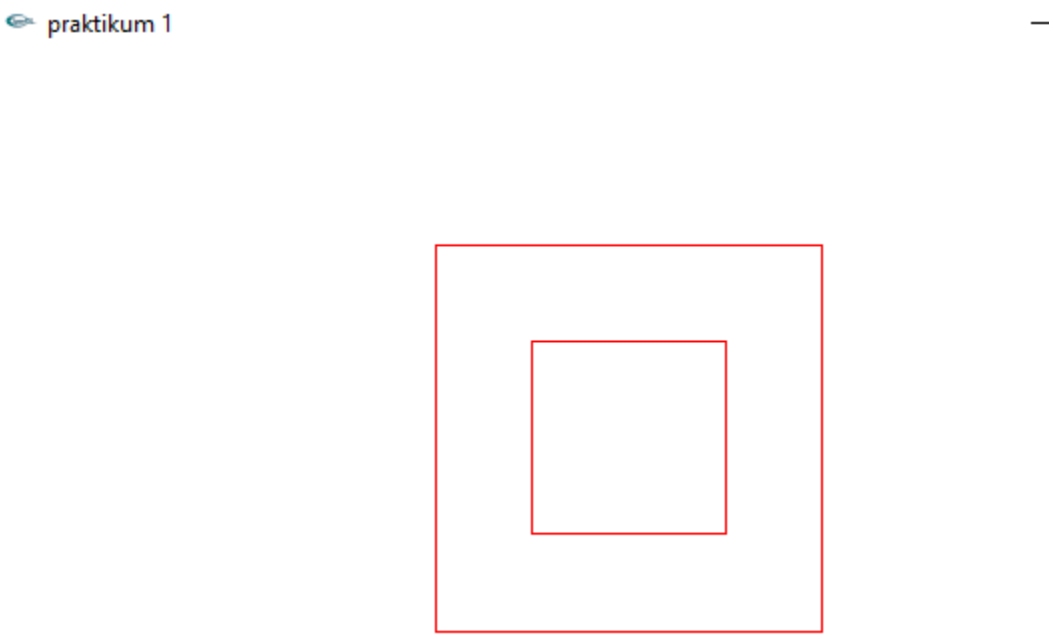
HURUF “O” | Ovaldy | 161111062

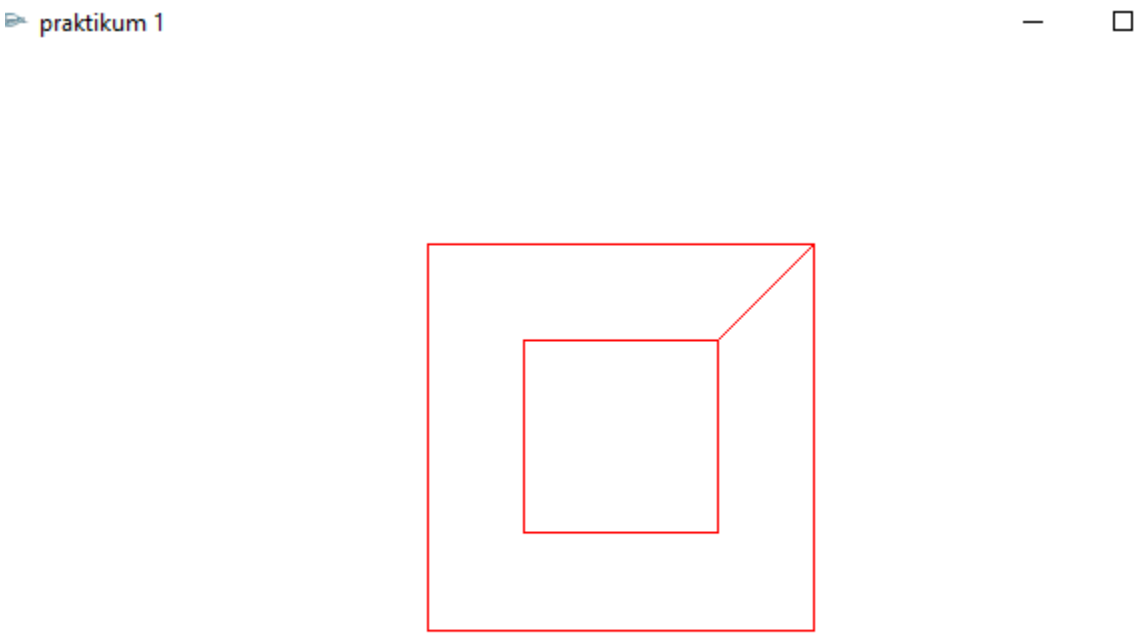
```
#include <windows.h>
#include "GL/glut.h"
#include "math.h"


void initGL()
{
  glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
  glClearDepth(1.0f);
  glEnable(GL_DEPTH_TEST);
  glDepthFunc(GL_LEQUAL);
  glShadeModel(GL_SMOOTH);
  glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST);
}

void timer(int value)
{
  glutPostRedisplay();
  glutTimerFunc(15, timer, 0);
}

void reshape(GLsizei width, GLsizei height)
{
  if (height == 0)
    height = 1;
  GLfloat aspect = (GLfloat)width / (GLfloat)height;
  glViewport(0, 0, width, height);
  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  gluPerspective(45.0f, aspect, 0.1f, 100.0f);
}

void display()
{
  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
  glMatrixMode(GL_MODELVIEW);
  glLoadIdentity();

  glTranslatef(0.0, 0.0, -6.0);
  glBegin(GL_LINES);
      glColor3f(1,0,0);// red
      glVertex3f(1,1.5,0);
      glVertex3f(-1,1.5,0);

      glVertex3f(-1,1.5,0);
      glVertex3f(-1,-0.5,0);

      glVertex3f(-1,-0.5,0);
      glVertex3f(1,-0.5,0);

      glVertex3f(1,-0.5,0);
      glVertex3f(1,1.5,0);

      glVertex3f(0.5,1,0);
      glVertex3f(-0.5,1,0);

      glVertex3f(-0.5,1,0);
      glVertex3f(-0.5,0,0);
```

```
59.
60.       glVertex3f(-0.5,0,0);
61.       glVertex3f(0.5,0,0);
62.
63.       glVertex3f(0.5,0,0);
64.       glVertex3f(0.5,1,0);
65.   glEnd();
66.
67.   glFlush();
68.   glutSwapBuffers();
69. }
70.
71. int main(int argc, char **argv)
72. {
73.   glutInit(&argc, argv);
74.   glutInitDisplayMode(GLUT_DOUBLE | GLUT_DEPTH);
75.   glutInitWindowSize(640, 480);
76.   glutInitWindowPosition(50, 50);
77.   glutCreateWindow("praktikum 1");
78.   glutDisplayFunc(display);
79.   glutReshapeFunc(reshape);
80.   initGL();
81.   glutTimerFunc(0, timer, 0);
82.   glutMainLoop();
83.   return 0;
84. }
```



GL LINE STRIP “O”

```
1.  glBegin(GL_LINE_STRIP);
2.        glColor3f(1,0,0);// red
3.        glVertex3f(1,1.5,0);
4.        glVertex3f(-1,1.5,0);
5.        glVertex3f(-1,-0.5,0);
6.        glVertex3f(1,-0.5,0);
7.        glVertex3f(1,1.5,0);
8.
9.        glVertex3f(0.5,1,0);
10.       glVertex3f(-0.5,1,0);
11.       glVertex3f(-0.5,0,0);
12.       glVertex3f(0.5,0,0);
13.       glVertex3f(0.5,1,0);
14.   glEnd();
```

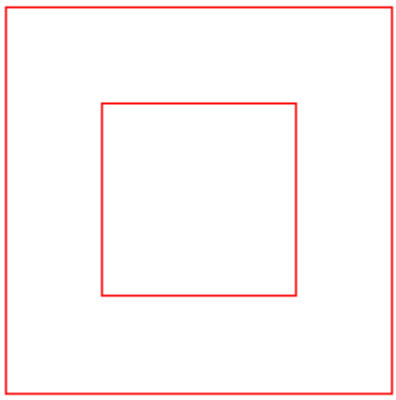


GL Line Loop “O”

```
1.  glBegin(GL_LINE_LOOP);
2.        glColor3f(1,0,0);// red
3.        glVertex3f(1,1.5,0);
4.        glVertex3f(-1,1.5,0);
5.        glVertex3f(-1,-0.5,0);
6.        glVertex3f(1,-0.5,0);
7.
8.
```

```
9.    glEnd();
10.   glBegin(GL_LINE_LOOP);
11.   glColor3f(1,0,0);// red
12.   glVertex3f(0.5,1,0);
13.       glVertex3f(-0.5,1,0);
14.       glVertex3f(-0.5,0,0);
15.       glVertex3f(0.5,0,0);
16.       glVertex3f(0.5,1,0);
17.
18.   glEnd();
```

praktikum 1



Perbedaan GL_LINE, GL_LINE_STRIP, GL_LINE_LOOP

```
1.  glTranslatef(0.0, 0.0, -6.0);
2.    glBegin(GL_LINES);
3.        glColor3f(1,0,0);// red
4.        glVertex3f(1,1.5,0);
5.        glVertex3f(-1,1.5,0);
6.
7.        glVertex3f(-1,1.5,0);
8.        glVertex3f(-1,-0.5,0);
9.
10.       glVertex3f(-1,-0.5,0);
11.       glVertex3f(1,-0.5,0);
12.
13.       glVertex3f(1,-0.5,0);
14.       glVertex3f(1,1.5,0);
15.  //HARUS MEMASANGKAN 2 TITIK
16.   glEnd();
17.   glTranslatef(0.0, 0.0, 0.0);
```

```
18.   glBegin(GL_LINE_LOOP);
19.   glColor3f(1,0,1);// red
20.       glVertex3f(0.5,1,0);
21.       glVertex3f(-0.5,1,0);
22.       glVertex3f(-0.5,0,0);
23.       glVertex3f(0.5,0,0);
24.   //LANGSUNG MEMBUAT TITIK YANG AKAN DIHUBUNGKAN SECARA LANGSUNG DAN YANG TERAKHIR AKAN
    MENGARAH KE TITIK PERTAMA SECARA OTOMATIS
25.   glEnd();
26.   glTranslatef(0.0, 0.3, 1.0);
27.   glBegin(GL_LINE_STRIP);
28.   glColor3f(0,0,1);// red
29.       glVertex3f(0.5,1,0);
30.       glVertex3f(-0.5,1,0);
31.       glVertex3f(-0.5,0,0);
32.       glVertex3f(0.5,0,0);
33.       glVertex3f(0.5,1,0);
34.   //LANGSUNG MEMBUAT TITIK YANG AKAN DIHUBUNGKAN SECARA LANGSUNG DAN YANG TERAKHIR
    HARUS DIARAHKAN KE TITIK PERTAMA SECARA MANUAL
35.
36.   glEnd();
```