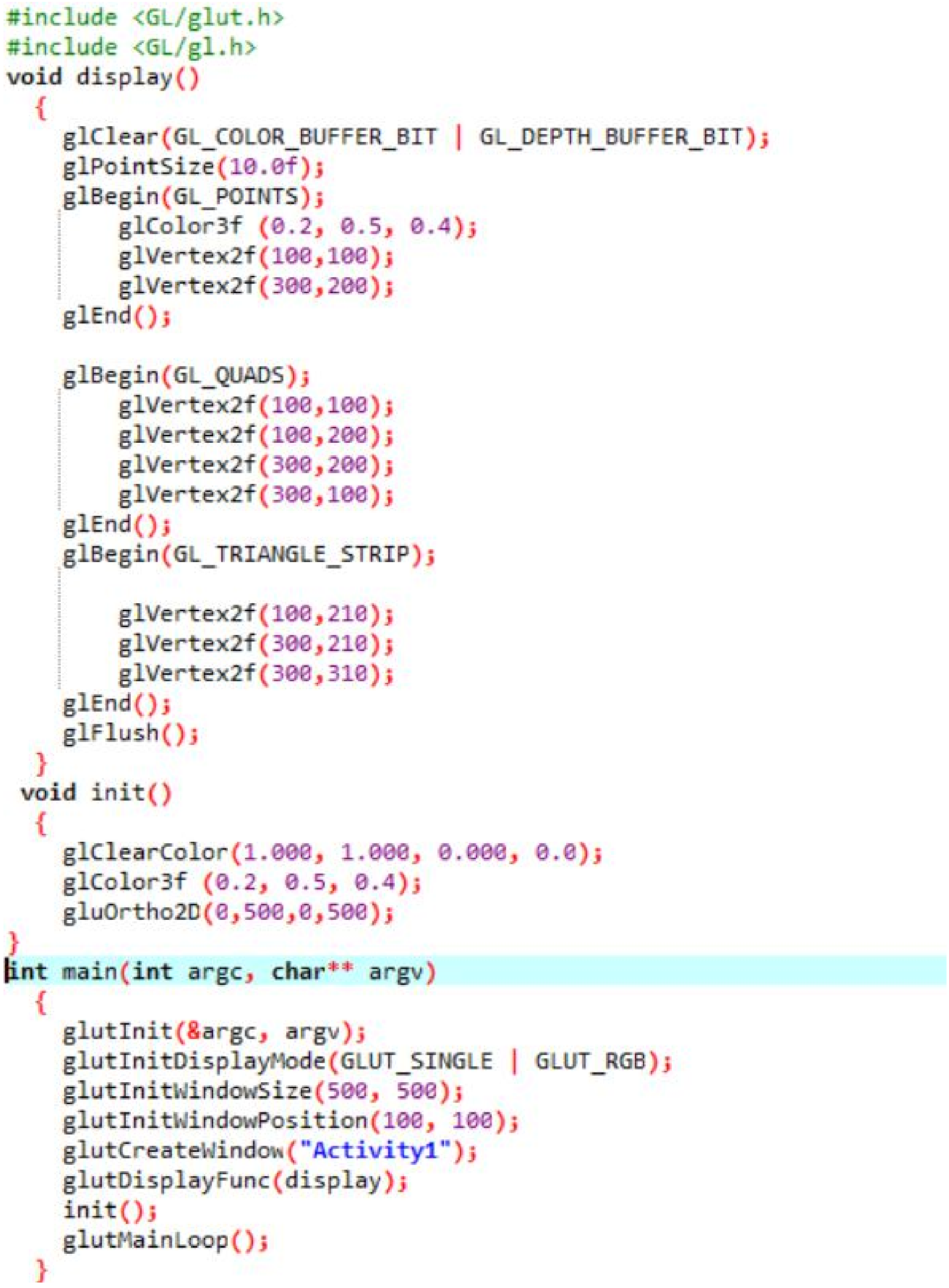
IS 2107

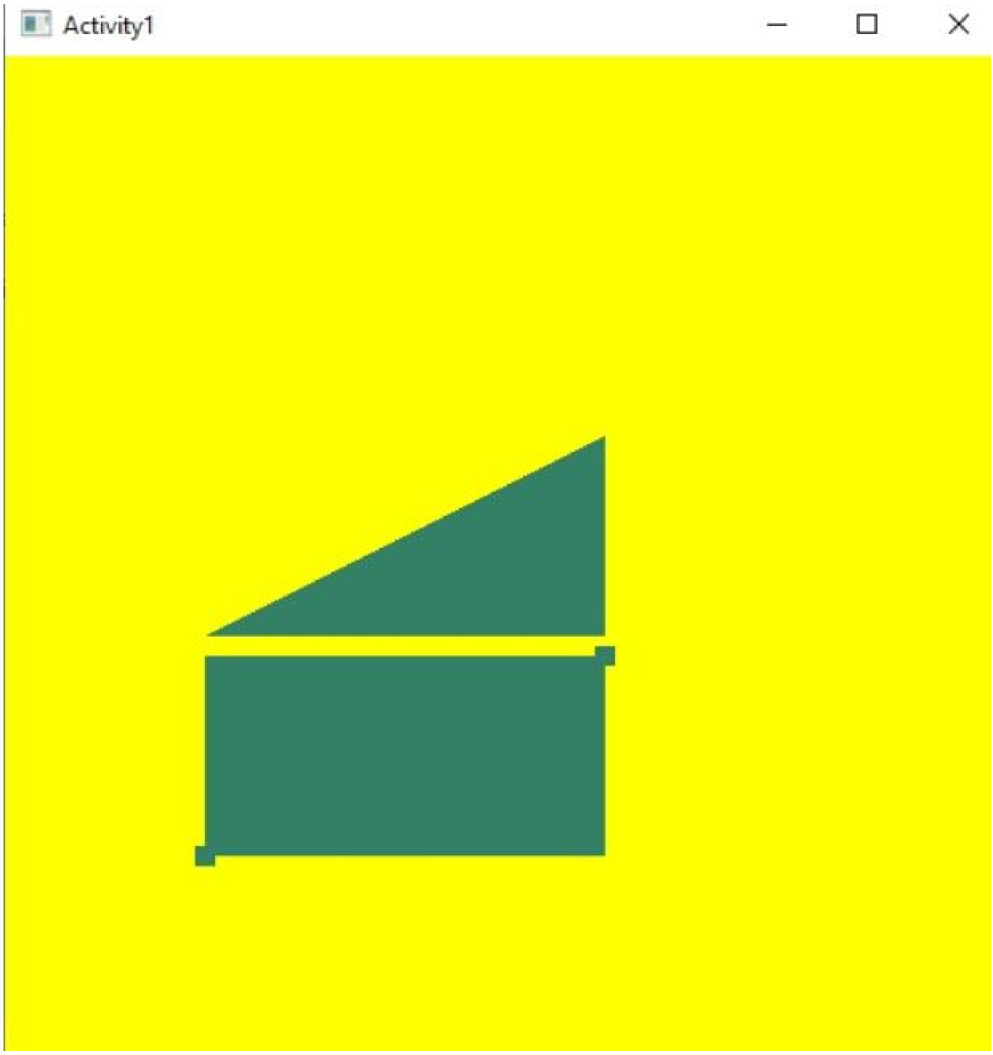
Graphics and Visualization

Practical Introduction to OpenGL

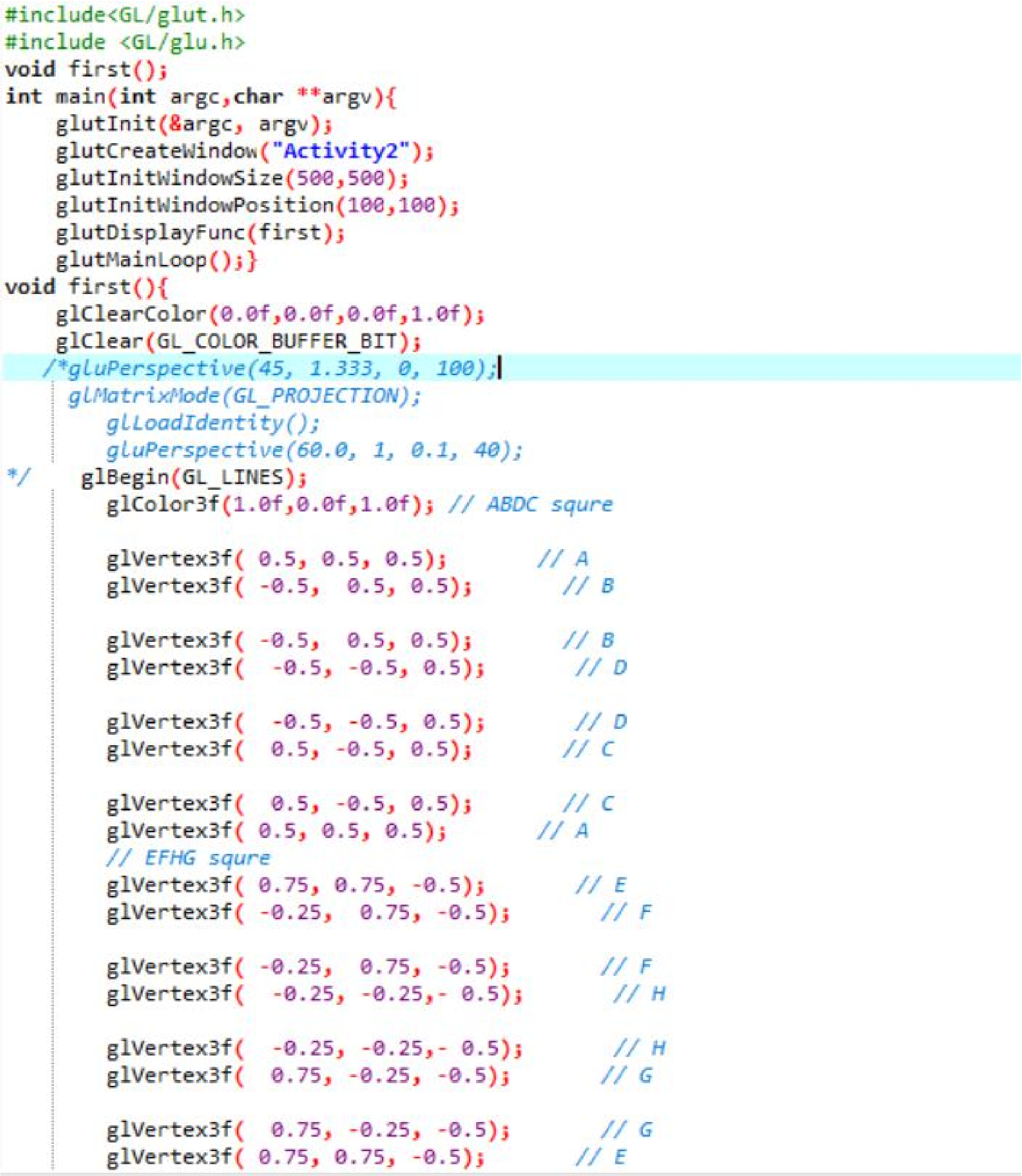
Activity 1



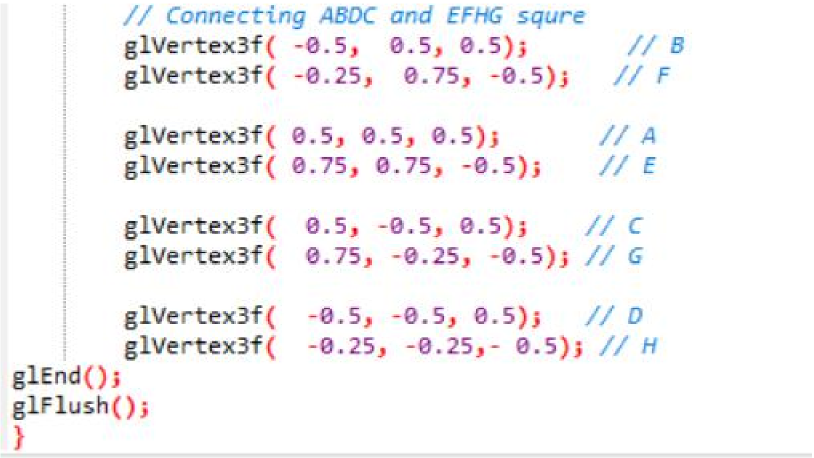
```
#include <GL/glut.h>
#include <GL/gl.h>
void display()
  {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glPointSize(10.0f);
    glBegin(GL_POINTS);
        glColor3f (0.2, 0.5, 0.4);
        glVertex2f(100,100);
        glVertex2f(300,200);
    glEnd();

    glBegin(GL_QUADS);
        glVertex2f(100,100);
        glVertex2f(100,200);
        glVertex2f(300,200);
        glVertex2f(300,100);
    glEnd();
    glBegin(GL_TRIANGLE_STRIP);

        glVertex2f(100,210);
        glVertex2f(300,210);
        glVertex2f(300,310);
    glEnd();
    glFlush();
  }
 void init()
  {
    glClearColor(1.000, 1.000, 0.000, 0.0);
    glColor3f (0.2, 0.5, 0.4);
    gluOrtho2D(0,500,0,500);
}
int main(int argc, char** argv)
  {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Activity1");
    glutDisplayFunc(display);
    init();
    glutMainLoop();
  }
```


Activity1

Activity 2



```
#include<GL/glut.h>
#include <GL/glu.h>
void first();
int main(int argc,char **argv){
    glutInit(&argc, argv);
    glutCreateWindow("Activity2");
    glutInitWindowSize(500,500);
    glutInitWindowPosition(100,100);
    glutDisplayFunc(first);
    glutMainLoop();}
void first(){
    glClearColor(0.0f,0.0f,0.0f,1.0f);
    glClear(GL_COLOR_BUFFER_BIT);
   /*gluPerspective(45, 1.333, 0, 100);
     glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        gluPerspective(60.0, 1, 0.1, 40);
*/    glBegin(GL_LINES);
        glColor3f(1.0f,0.0f,1.0f); // ABDC squre

        glVertex3f( 0.5, 0.5, 0.5);       // A
        glVertex3f( -0.5,  0.5, 0.5);       // B

        glVertex3f( -0.5,  0.5, 0.5);       // B
        glVertex3f(  -0.5, -0.5, 0.5);       // D

        glVertex3f(  -0.5, -0.5, 0.5);       // D
        glVertex3f(  0.5, -0.5, 0.5);       // C

        glVertex3f(  0.5, -0.5, 0.5);       // C
        glVertex3f( 0.5, 0.5, 0.5);       // A
        // EFHG squre
        glVertex3f( 0.75, 0.75, -0.5);       // E
        glVertex3f( -0.25,  0.75, -0.5);       // F

        glVertex3f( -0.25,  0.75, -0.5);       // F
        glVertex3f(  -0.25, -0.25,- 0.5);       // H

        glVertex3f(  -0.25, -0.25,- 0.5);       // H
        glVertex3f(  0.75, -0.25, -0.5);       // G

        glVertex3f(  0.75, -0.25, -0.5);       // G
        glVertex3f( 0.75, 0.75, -0.5);       // E
```



```
        // Connecting ABDC and EFHG squre
        glVertex3f( -0.5,  0.5, 0.5);       // B
        glVertex3f( -0.25,  0.75, -0.5);   // F

        glVertex3f( 0.5, 0.5, 0.5);       // A
        glVertex3f( 0.75, 0.75, -0.5);    // E

        glVertex3f(  0.5, -0.5, 0.5);    // C
        glVertex3f(  0.75, -0.25, -0.5); // G

        glVertex3f(  -0.5, -0.5, 0.5);   // D
        glVertex3f(  -0.25, -0.25,- 0.5); // H
glEnd();
glFlush();
}
```