

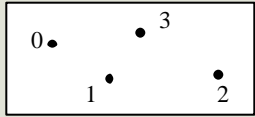
OpenGL Quick Reference #1

Introduction to Drawing in OpenGL

Simple OpenGL drawing code typically looks like this:

```

gl Color*( color ) ;
gl Begin( geometric primitive ) ;
    gl Vertex*( . . . ) ; // vertex 0
    gl Vertex*( . . . ) ; // vertex 1
    ...
    gl Vertex*( . . . ) ; // vertex n-1
gl End() ;
    
```



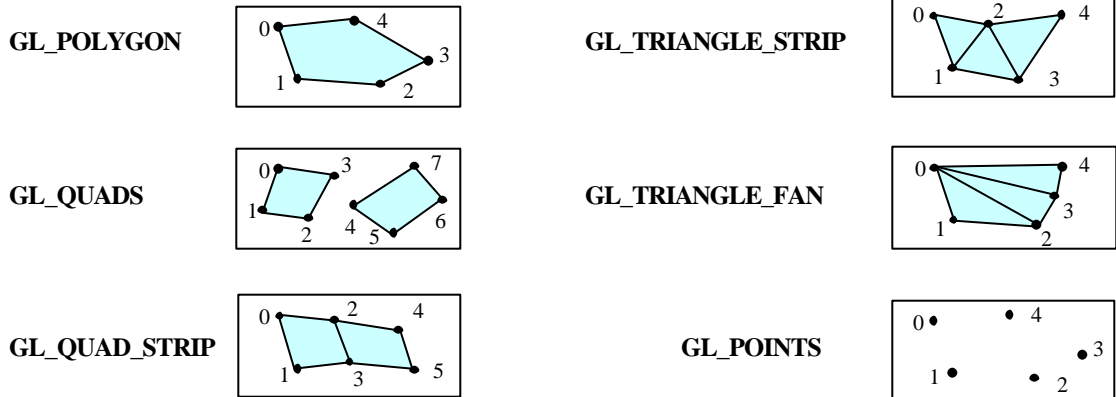
A. Specifying the color [red,green,blue levels]

```

void glColor3d( double r, double g, double b ) ; // 0.0 £ r,g,b £ 1.0
void glColor3dv( const double rgb[] ) ; // rgb[0,1,2]=r,g,b ; 0.0 £ r,g,b £ 1.0
    
```

Note: You can also change colors in between calls to glVertex*(...) .

B. Specifying the geometric primitive



C. Specifying the vertices

```

void glVertex3d( double x, double y, double z ) ; // x, y, z coordinates in 3D (y up, z out by default)
void glVertex3dv( const double v[] ) ; // v[0]=x, v[1]=y, v[2]=z coordinates in 3D
    
```

Instead of making calls to glVertex*() for each vertex, you can also pile them into an array of 3D double vertices (call the array vtx[] here):

```

glEnableClientState( GL_VERTEX_ARRAY ) ;
glVertexPointer( 3, GL_DOUBLE, 0, vtx ) ; // the 3 means vertices are 3D (three doubles each)
glDrawArrays( GL_POLYGON, 0, numVertices ) ; // the array should then have 3*numVertices entries
glDisableClientState( GL_VERTEX_ARRAY ) ;
    
```

In C++, use `vector<double> vtx`, passing `&(vtx[0])` to these methods instead of `vtx`.