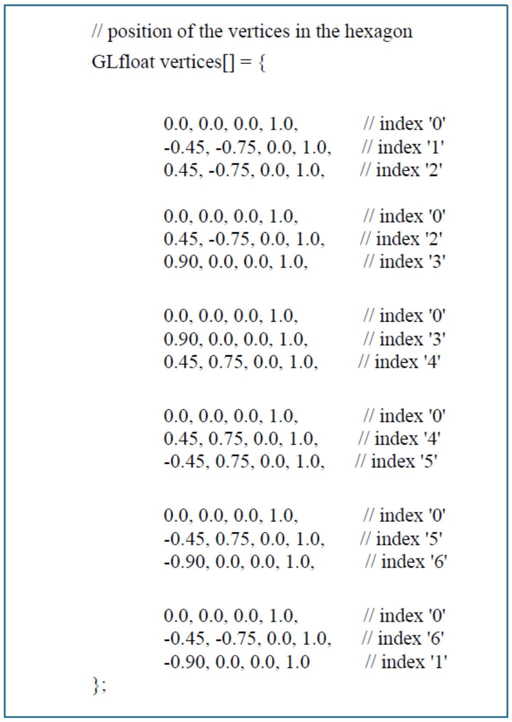
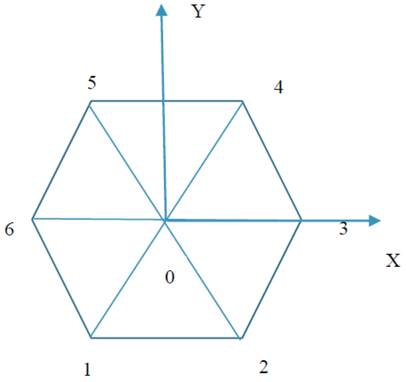
Tutorial 1 Week 1

1. Unzip the uploaded file **Recangle.zip**. Next, compile and run the program. Take a look at ‘**display**’ function. The first parameter in ‘**glDrawArrays**’ function means ‘**mode**’.
(a)Try to change the mode from **GL_TRIANGLES** to **GL_POINTS**, **GL_LINES**, **GL_LINE_STRIP**, **GL_LINE_LOOP**, **GL_TRIANGLE_FAN** etc.
(b)Take a look at the callback function Keyboard. See how you can switch between polygon and wireframe mode.

2. Draw one hexagon with the following set of vertices:



```
// position of the vertices in the hexagon
GLfloat vertices[] = {

          0.0, 0.0, 0.0, 1.0,          // index '0'
          -0.45, -0.75, 0.0, 1.0,      // index '1'
          0.45, -0.75, 0.0, 1.0,       // index '2'

          0.0, 0.0, 0.0, 1.0,          // index '0'
          0.45, -0.75, 0.0, 1.0,       // index '2'
          0.90, 0.0, 0.0, 1.0,         // index '3'

          0.0, 0.0, 0.0, 1.0,          // index '0'
          0.90, 0.0, 0.0, 1.0,         // index '3'
          0.45, 0.75, 0.0, 1.0,        // index '4'

          0.0, 0.0, 0.0, 1.0,          // index '0'
          0.45, 0.75, 0.0, 1.0,        // index '4'
          -0.45, 0.75, 0.0, 1.0,       // index '5'

          0.0, 0.0, 0.0, 1.0,          // index '0'
          -0.45, 0.75, 0.0, 1.0,       // index '5'
          -0.90, 0.0, 0.0, 1.0,        // index '6'

          0.0, 0.0, 0.0, 1.0,          // index '0'
          -0.45, -0.75, 0.0, 1.0,      // index '6'
          -0.90, 0.0, 0.0, 1.0         // index '1'
};
```



a. What do you observe? There’s a lot of repetitions of vertices in the vertex array. Models can be very large with millions of vertices and your vertex array will occupy a lot of memory. How will you address this problem?

b. Let’s use glDrawElements instead of glDrawArrays. How does it work? Create one index array named **indices** of type int and add all the index numbers as shown to the right of each vertex of the hexagon.

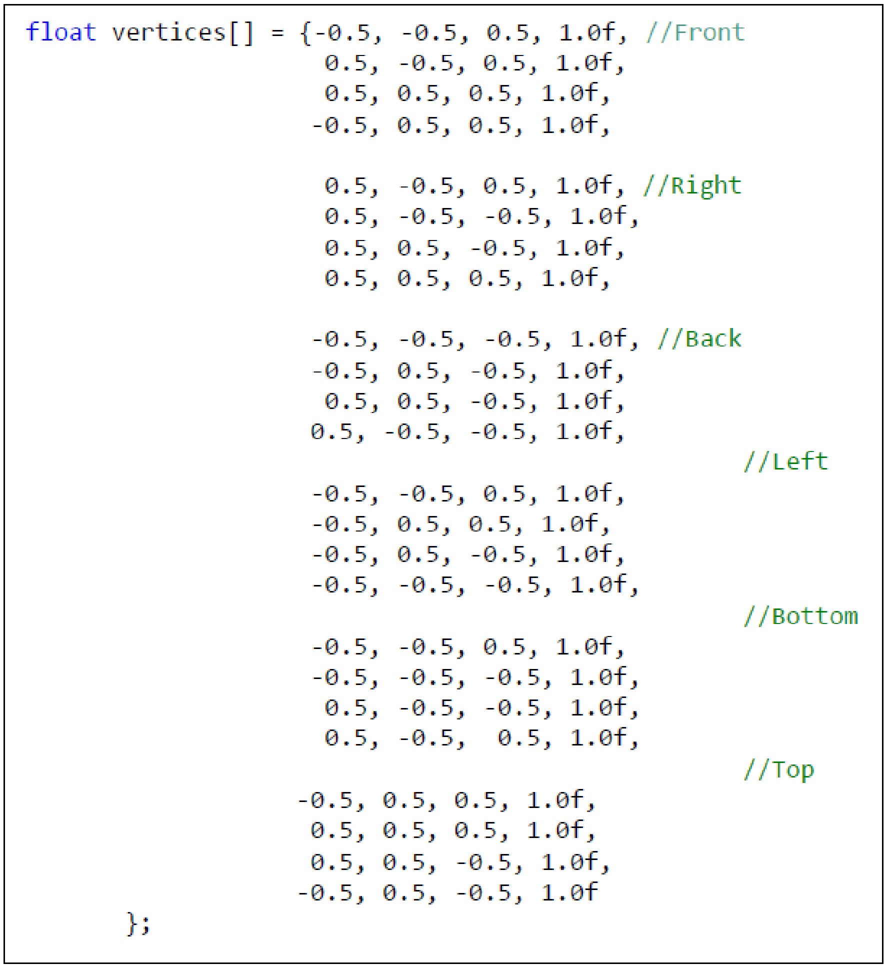
c. Attach another buffer to the vertex array object. Add the following lines:
glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, Buffers[1]); glBufferData(GL_ELEMENT_ARRAY_BUFFER, sizeof(indices), indices, GL_STATIC_DRAW);

d. Now use glDrawElements instead of glDrawArrays as follows:
glDrawElements(GL_TRIANGLES, NUM_INDICES, GL_UNSIGNED_INT, 0);

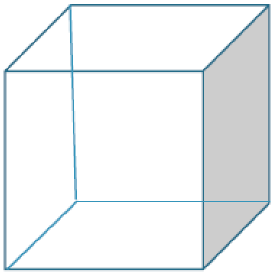
What will be the value of NUM_INDICES?

3. As the object becomes complex or you add a number of objects to the scene, the main file looks cluttered. Now unzip the file named **objects.zip**. Now you find there is one header file named “Triangle.h” and two source files “Triangle.cpp” and “main.cpp”. Make the program modular by creating separate files (.h and .cpp) for separate objects. C language has been used. Please feel free to use C++ if you’d like to.

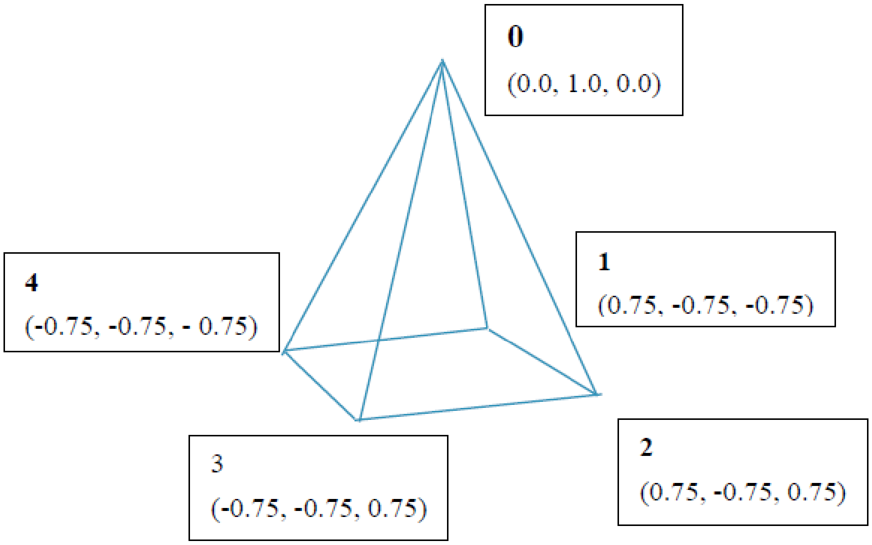
4. Now add a cube with the following vertex information. Add “**Cube.h**” and “**Cube.cpp**” files.



```
float vertices[] = {-0.5, -0.5, 0.5, 1.0f, //Front
                     0.5, -0.5, 0.5, 1.0f,
                     0.5, 0.5, 0.5, 1.0f,
                    -0.5, 0.5, 0.5, 1.0f,

                     0.5, -0.5, 0.5, 1.0f, //Right
                     0.5, -0.5, -0.5, 1.0f,
                     0.5, 0.5, -0.5, 1.0f,
                     0.5, 0.5, 0.5, 1.0f,

                    -0.5, -0.5, -0.5, 1.0f, //Back
                    -0.5, 0.5, -0.5, 1.0f,
                     0.5, 0.5, -0.5, 1.0f,
                    0.5, -0.5, -0.5, 1.0f,
                                                  //Left
                    -0.5, -0.5, 0.5, 1.0f,
                    -0.5, 0.5, 0.5, 1.0f,
                    -0.5, 0.5, -0.5, 1.0f,
                    -0.5, -0.5, -0.5, 1.0f,
                                                  //Bottom
                    -0.5, -0.5, 0.5, 1.0f,
                    -0.5, -0.5, -0.5, 1.0f,
                     0.5, -0.5, -0.5, 1.0f,
                     0.5, -0.5,  0.5, 1.0f,
                                                  //Top
                   -0.5, 0.5, 0.5, 1.0f,
                    0.5, 0.5, 0.5, 1.0f,
                    0.5, 0.5, -0.5, 1.0f,
                   -0.5, 0.5, -0.5, 1.0f
       };
```



5. Can you draw a **pyramid** with the following information: There will be four triangles (sides) and one rectangle (the base)


0
(0.0, 1.0, 0.0)
4
(-0.75, -0.75, - 0.75)
1
(0.75, -0.75, -0.75)
2
(0.75, -0.75, 0.75)
3
(-0.75, -0.75, 0.75)