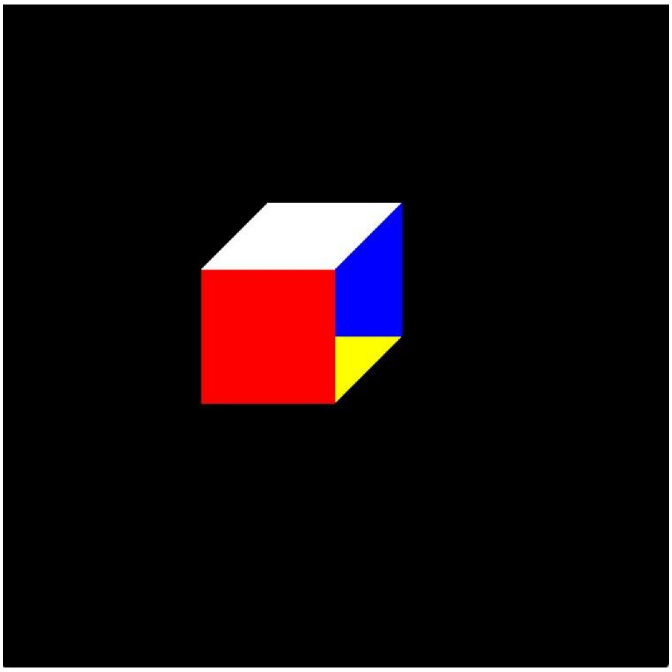
SE3032 – Graphics and Visualization Lab 04

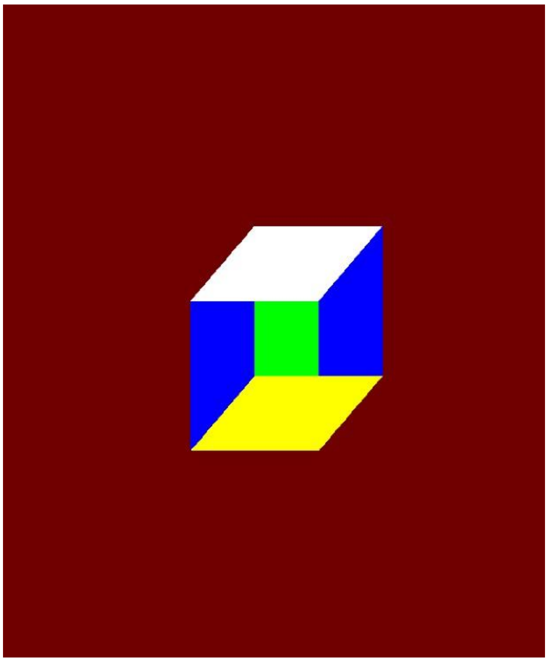
IT23226128 – D.M.D.G.B.Seneviratne

Activity 01

```
#define GL_SILENCE_DEPRECATION
#include <GLUT/glut.h>
#include <OpenGL/gl.h>
#include <OpenGL/glu.h>
#include <stdlib.h>

void createCube() {     //
Clear the color buffer
   glClear(GL_COLOR_BUFFER_BIT);

   // Back face (Green)
glColor3f(0.0, 1.0, 0.0);
   glBegin(GL_POLYGON);
   glVertex3f(-0.2, 0.0, -0.4);
glVertex3f(-0.2, 0.4, -0.4);
glVertex3f(0.2, 0.4, -0.4);
glVertex3f(0.2, 0.0, -0.4);
glEnd();

   // Left face (Blue)
glColor3f(0.0, 0.0, 1.0);
   glBegin(GL_POLYGON);
   glVertex3f(-0.4, -0.2, 0.0);
glVertex3f(-0.4, 0.2, 0.0);
glVertex3f(-0.2, 0.4, -0.4);
glVertex3f(-0.2, 0.0, -0.4);
glEnd();

   // Right face (Blue)
glColor3f(0.0, 0.0, 1.0);
   glBegin(GL_POLYGON);
glVertex3f(0.0, -0.2, 0.0);
glVertex3f(0.0, 0.2, 0.0);
glVertex3f(0.2, 0.4, -0.4);
glVertex3f(0.2, 0.0, -0.4);
glEnd();

   // Top face (White)
glColor3f(1.0, 1.0, 1.0);
   glBegin(GL_POLYGON);
   glVertex3f(-0.4, 0.2, 0.0);
glVertex3f(0.0, 0.2, 0.0);
glVertex3f(0.2, 0.4, -0.4);
glVertex3f(-0.2, 0.4, -0.4);
```

```
  glEnd();

   // Bottom face (Yellow)
glColor3f(1.0, 1.0, 0.0);
   glBegin(GL_POLYGON);
   glVertex3f(-0.4, -0.2, 0.0);
glVertex3f(0.0, -0.2, 0.0);
glVertex3f(0.2, 0.0, -0.4);
glVertex3f(-0.2, 0.0, -0.4);     glEnd();

   // Front face (Red)
glColor3f(1.0, 0.0, 0.0);
   glBegin(GL_POLYGON);
   glVertex3f(-0.4, -0.2, 0.0);
glVertex3f(-0.4, 0.2, 0.0);
glVertex3f(0.0, 0.2, 0.0);
glVertex3f(0.0, -0.2, 0.0);     glEnd();

   glFlush();  // Render the scene
}

void display(void) {     createCube();
}

int main(int argc, char** argv) {
glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitWindowPosition(500, 500);
glutInitWindowSize(500, 500);     glutCreateWindow("Cube
of IT23226128");

   glClearColor(0.0, 0.0, 0.0, 0.0);

   glutDisplayFunc(display);
glutMainLoop();     return 0;
}
```

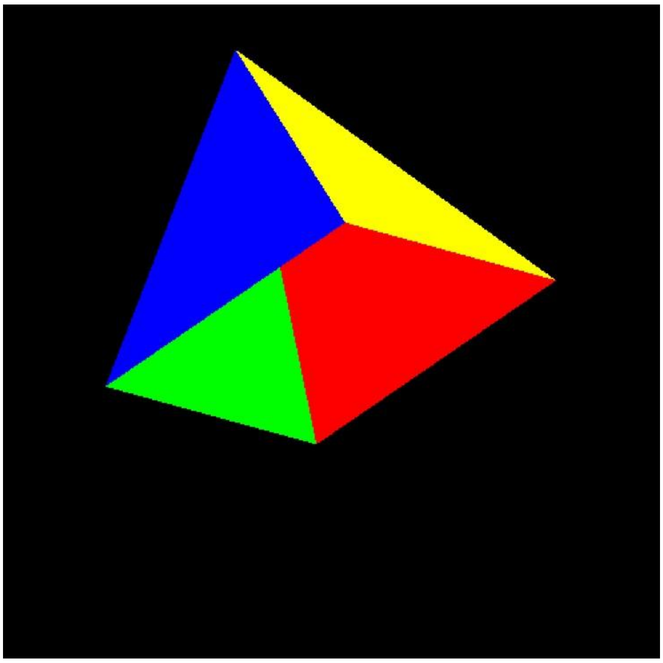




Activity 02

```
#define GL_SILENCE_DEPRECATION
#include <GLUT/glut.h>
#include <OpenGL/gl.h>
#include <OpenGL/glu.h>
#include <stdlib.h>

void createPyramid() {
glClear(GL_COLOR_BUFFER_BIT);

   // Base (Square - using two triangles)
glColor3f(0.5, 0.5, 0.5);

   // First triangle of base
   glBegin(GL_TRIANGLES);
   glVertex3f(-0.5, 0.0, -0.5);
glVertex3f(0.5, 0.0, -0.5);
glVertex3f(-0.5, 0.0, 0.5);
glEnd();

   // Second triangle of base
glBegin(GL_TRIANGLES);
   glVertex3f(0.5, 0.0, -0.5);
glVertex3f(0.5, 0.0, 0.5);
glVertex3f(-0.5, 0.0, 0.5);
glEnd();

   // Front face (Red)
glColor3f(1.0, 0.0, 0.0);
   glBegin(GL_TRIANGLES);
   glVertex3f(-0.5, 0.0, -0.5);
glVertex3f(0.5, 0.0, -0.5);
glVertex3f(0.0, 1.0, 0.0);     glEnd();

   // Right face (Green)
glColor3f(0.0, 1.0, 0.0);
   glBegin(GL_TRIANGLES);
   glVertex3f(0.5, 0.0, -0.5);
glVertex3f(0.5, 0.0, 0.5);
glVertex3f(0.0, 1.0, 0.0);     glEnd();

   // Back face (Blue)
glColor3f(0.0, 0.0, 1.0);
   glBegin(GL_TRIANGLES);
   glVertex3f(0.5, 0.0, 0.5);
glVertex3f(-0.5, 0.0, 0.5);
glVertex3f(0.0, 1.0, 0.0);     glEnd();

   // Left face (Yellow)
   glColor3f(1.0, 1.0, 0.0);
  glBegin(GL_TRIANGLES);
```

```
   glVertex3f(-0.5, 0.0, 0.5);
glVertex3f(-0.5, 0.0, -0.5);
glVertex3f(0.0, 1.0, 0.0);     glEnd();

   glFlush();
}

void display(void) {
createPyramid();
}

void keyboard(unsigned char key, int x, int y) {
switch (key) {        case 'r': case 'R':
          glRotatef(10.0, 0.0, 1.0, 0.0); // Rotate around Y-axis
break;        case 'x': case 'X':
          glRotatef(10.0, 1.0, 0.0, 0.0); // Rotate around X-axis
break;        case 'q': case 'Q':
          exit(0);
break;
   }
   glutPostRedisplay();
}

int main(int argc, char** argv) {
glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitWindowPosition(200, 200);     glutInitWindowSize(500, 500);
glutCreateWindow("pyramid – IT23226128");

   glClearColor(0.0, 0.0, 0.0, 1.0);

   glutKeyboardFunc(keyboard);
glutDisplayFunc(display);
glutMainLoop();     return 0;
}
```



Assignment

```
#define GL_SILENCE_DEPRECATION
#include <GLUT/glut.h>
#include <stdlib.h>

static const float APEX_PUSH_X = 0.40f;

static void init(void) {
glClearColor(0.0, 0.0, 0.0, 1.0);
glEnable(GL_DEPTH_TEST);
glDepthFunc(GL_LEQUAL);
glClearDepth(1.0);
}

static void reshape(int w, int h) {
glViewport(0, 0, w, h);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();     glOrtho(-1.0, 1.0,   -1.0, 1.0,   -
1.0, 1.0); // z in [-1,1]
   glMatrixMode(GL_MODELVIEW);
   glLoadIdentity();
}

static void drawScene(void) {
   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

   // Back face (Green)
glColor3f(0.0, 1.0, 0.0);
   glBegin(GL_POLYGON);
      glVertex3f(-0.2f, 0.0f, -0.4f);
glVertex3f(-0.2f, 0.4f, -0.4f);
glVertex3f( 0.2f, 0.4f, -0.4f);
glVertex3f( 0.2f, 0.0f, -0.4f);
glEnd();

   // Left face (Blue)
glColor3f(0.0, 0.0, 1.0);
   glBegin(GL_POLYGON);
      glVertex3f(-0.4f, -0.2f,  0.0f);
glVertex3f(-0.4f,  0.2f,  0.0f);
glVertex3f(-0.2f,  0.4f, -0.4f);
glVertex3f(-0.2f,  0.0f, -0.4f);
glEnd();

   // Top face (White)
glColor3f(1.0, 1.0, 1.0);
   glBegin(GL_POLYGON);
      glVertex3f(-0.4f, 0.2f,   0.0f);
glVertex3f(   0.0f,   0.2f,      0.0f);
glVertex3f(    0.2f,    0.4f,    -0.4f);
glVertex3f(-0.2f,     0.4f,     -0.4f);
glEnd();
```

```
   // Bottom face (Yellow)
glColor3f(1.0, 1.0, 0.0);
   glBegin(GL_POLYGON);
      glVertex3f(-0.4f, -0.2f,  0.0f);
glVertex3f(   0.0f,   -0.2f,      0.0f);
glVertex3f(   0.2f,      0.0f,   -0.4f);
glVertex3f(-0.2f,       0.0f,    -0.4f);
glEnd();

   // Front face (Red)
glColor3f(1.0, 0.0, 0.0);
   glBegin(GL_POLYGON);
      glVertex3f(-0.4f, -0.2f, 0.0f);
glVertex3f(-0.4f,        0.2f,    0.0f);
glVertex3f(   0.0f,       0.2f,   0.0f);
glVertex3f(    0.0f,    -0.2f,    0.0f);
glEnd();


   const GLfloat B1[3] = { 0.0f, -0.2f,  0.0f };  // front-bottom-right
const GLfloat B2[3] = { 0.0f,  0.2f,  0.0f };  // front-top-right     const
GLfloat B3[3] = { 0.2f,  0.4f, -0.4f };  // back-top-right     const
GLfloat B4[3] = { 0.2f,  0.0f, -0.4f };  // back-bottom-right


   const float cx = (B1[0] + B2[0] + B3[0] + B4[0]) * 0.25f; // ~0.1
const float cy = (B1[1] + B2[1] + B3[1] + B4[1]) * 0.25f; // ~0.1     const
float cz = (B1[2] + B2[2] + B3[2] + B4[2]) * 0.25f; // ~-0.2

   // Apex pushed outward along +X (to the "right")
const GLfloat A[3] = { cx + APEX_PUSH_X, cy, cz };

   // Four triangular sides around the base
glColor3f(1.0f, 1.0f, 0.0f); // Yellow
   glBegin(GL_TRIANGLES); glVertex3fv(B1); glVertex3fv(B2); glVertex3fv(A); glEnd();

   glColor3f(0.0f, 1.0f, 1.0f); // Cyan
   glBegin(GL_TRIANGLES); glVertex3fv(B2); glVertex3fv(B3); glVertex3fv(A); glEnd();

   glColor3f(1.0f, 0.5f, 0.0f); // Orange
   glBegin(GL_TRIANGLES); glVertex3fv(B3); glVertex3fv(B4); glVertex3fv(A); glEnd();

   glColor3f(0.6f, 0.2f, 1.0f); // Purple     glBegin(GL_TRIANGLES); glVertex3fv(B4);
glVertex3fv(B1); glVertex3fv(A); glEnd();

   glutSwapBuffers();
}

int main(int argc, char** argv) {
glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH); // depth + double buffer
glutInitWindowPosition(200, 200);
```

```
   glutInitWindowSize(500, 500);     glutCreateWindow("Cube +
Right-Side Pyramid (IT23226128)");

init();
   glutReshapeFunc(reshape);
glutDisplayFunc(drawScene);
glutMainLoop();     return 0;
}
```