Experiment – 10 : Construct the following 3d shapes: Cube And Sphere

~~~~~~

Ques : Construct the 3d shape Cube

Code : -

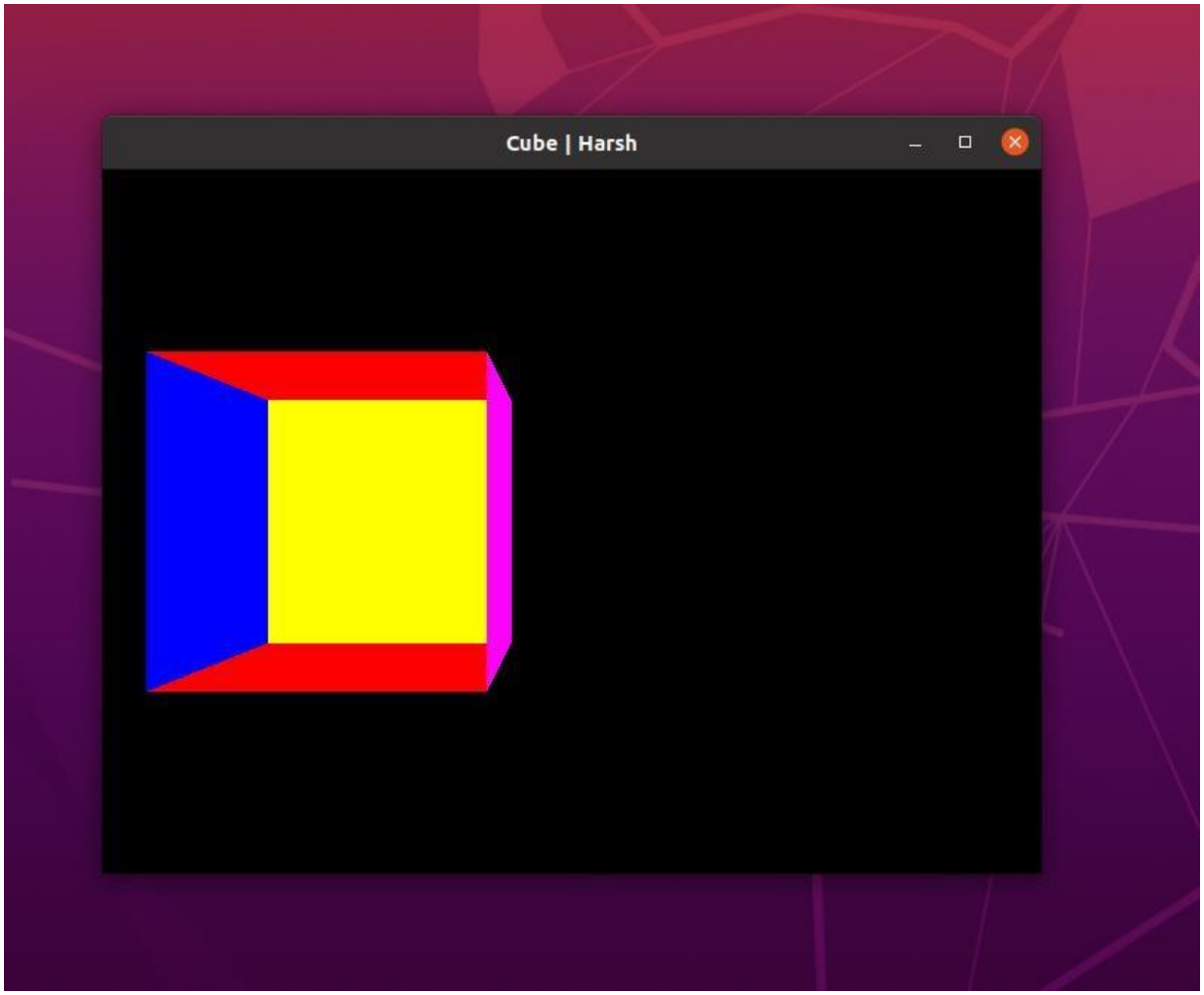
```
#include<GL/glut.h>
#include<GL/gl.h>
char title[] = "3D Shapes";
void initGL() {
  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
  glClearDepth(1.0f);
  glEnable(GL_DEPTH_TEST);
  glDepthFunc(GL_LEQUAL);
  glShadeModel(GL_SMOOTH);
  glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST);
}
void display() {
  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // Clear color and depth buffers
  glMatrixMode(GL_MODELVIEW);     // To operate on model-view matrix

  // Render a color-cube consisting of 6 quads with different colors
  glLoadIdentity();                 // Reset the model-view matrix
  glTranslatef(-1.5f, 0.0f, -6.0f); // Move right and into the screen
```
~~~~~~

```
  glBegin(GL_QUADS);              // Begin drawing the color cube with 6 quads
    // Top face (y = 1.0f)
    // Define vertices in counter-clockwise (CCW) order with normal pointing out
    glColor3f(0.0f, 1.0f, 0.0f);     // Green
    glVertex3f( 1.0f, 1.0f, -1.0f);
    glVertex3f(-1.0f, 1.0f, -1.0f);
    glVertex3f(-1.0f, 1.0f, 1.0f);
    glVertex3f( 1.0f, 1.0f,  1.0f);
    // Bottom face (y = -1.0f)
    glColor3f(1.0f, 0.5f, 0.0f);     // Orange
    glVertex3f( 1.0f, -1.0f,  1.0f);
    glVertex3f(-1.0f, -1.0f,  1.0f);
    glVertex3f(-1.0f, -1.0f, -1.0f);
    glVertex3f( 1.0f, -1.0f, -1.0f);
    // Front face (z = 1.0f)
    glColor3f(1.0f, 0.0f, 0.0f);     // Red
    glVertex3f( 1.0f, 1.0f, 1.0f);
    glVertex3f(-1.0f, 1.0f, 1.0f);
    glVertex3f(-1.0f, -1.0f, 1.0f);
    glVertex3f( 1.0f, -1.0f, 1.0f);
    // Back face (z = -1.0f)
    glColor3f(1.0f, 1.0f, 0.0f);     // Yellow
```

```
      glVertex3f( 1.0f, -1.0f, -1.0f);
      glVertex3f(-1.0f, -1.0f, -1.0f);
      glVertex3f(-1.0f,  1.0f, -1.0f);
      glVertex3f( 1.0f,  1.0f, -1.0f);
      // Left face (x = -1.0f)
      glColor3f(0.0f, 0.0f, 1.0f);     // Blue
      glVertex3f(-1.0f, 1.0f,  1.0f);
      glVertex3f(-1.0f,  1.0f, -1.0f);
      glVertex3f(-1.0f, -1.0f, -1.0f);
      glVertex3f(-1.0f, -1.0f,  1.0f);
      // Right face (x = 1.0f)
      glColor3f(1.0f, 0.0f, 1.0f);     // Magenta
      glVertex3f(1.0f, 1.0f, -1.0f);
      glVertex3f(1.0f, 1.0f, 1.0f);
      glVertex3f(1.0f, -1.0f,  1.0f);
      glVertex3f(1.0f, -1.0f, -1.0f);
   glEnd();
glutSwapBuffers();
}
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);
}
int main(int argc, char** argv) {
  glutInit(&argc, argv);
  glutInitDisplayMode(GLUT_DOUBLE);
  glutInitWindowSize(640, 480);
  glutInitWindowPosition(50, 50);
  glutCreateWindow("Cube | Harsh");
  glutDisplayFunc(display);
  glutReshapeFunc(reshape);
  initGL();
  glutMainLoop();
  return 0;
}
```

- **Output Are As Follows : -**



Ques : Construct the 3d shape Sphere

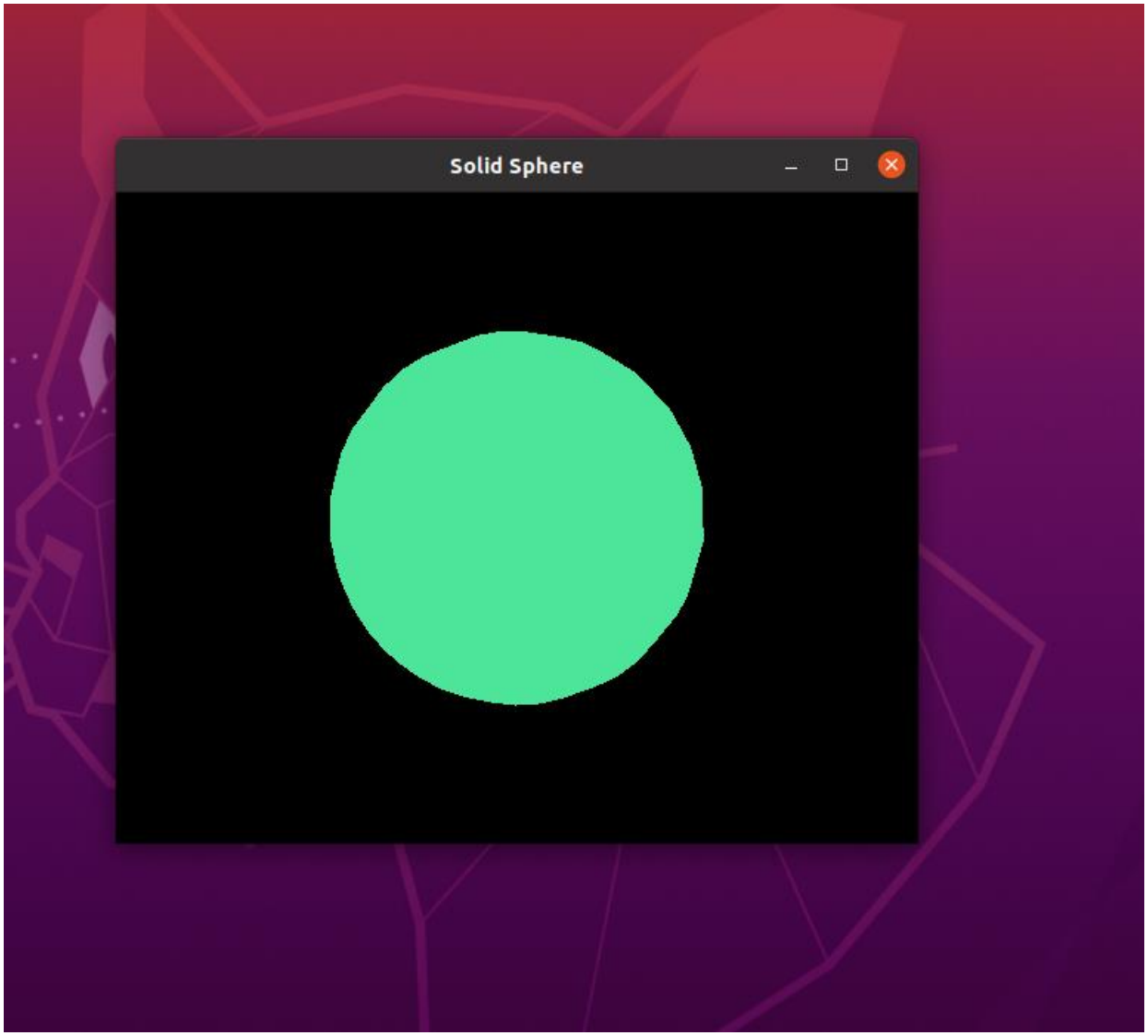
Code:

```
#include <GL/glut.h>
GLfloat xRotated, yRotated, zRotated;
GLdouble radius=1;


void display(void);
void reshape(int x, int y);
void idle(void)
{

  xRotated += 0.01;

   zRotated += 0.01;
   display();
}
```

```
int main (int argc, char **argv)
{
   glutInit(&argc, argv);
   glutInitWindowSize(350,350);
   glutCreateWindow("Solid Sphere");
   xRotated = yRotated = zRotated = 30.0;
   xRotated=43;
   yRotated=50;

   glutDisplayFunc(display);
   glutReshapeFunc(reshape);
  glutIdleFunc(idle);
   glutMainLoop();
   return 0;
}

void display(void)
{

   glMatrixMode(GL_MODELVIEW);

   glClear(GL_COLOR_BUFFER_BIT);
   glLoadIdentity();

   glTranslatef(0.0,0.0,-5.0);

   glColor3f(0.9, 0.3, 0.2);

   glRotatef(xRotated,1.0,0.0,0.0);

   glRotatef(yRotated,0.0,1.0,0.0);

   glRotatef(zRotated,0.0,0.0,1.0);

   glScalef(1.0,1.0,1.0);
```

```
    // built-in (glut library) function , draw you a sphere.
    glutSolidSphere(radius,20,20);
    // Flush buffers to screen

    glFlush();

}

void reshape(int x, int y)
{
    if (y == 0 || x == 0) return;
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(39.0,(GLdouble)x/(GLdouble)y,0.6,21.0);
    glMatrixMode(GL_MODELVIEW);
    glViewport(0,0,x,y);
}
```

- Output Are As Follows : -


Solid Sphere