Aim:
Program, using OpenGL functions, to draw a simple shaded scene consisting of a tea pot on a table. Define suitably the position and properties of the light source along with the properties of the properties of the surfaces of the solid object used in the scene.

Algorithm:

1. Use function glutSolidCube() to draw wall and table
2. Use the same function to draw 4 cubes and then scale it in downward direction to make it look like tablelegs
3. glutSolidTeapot() is used to draw teapot

Code:

```
     //teapot.c
        #include<stdio.h>
        #include<GL/glut.h>
        void wall(double thickness)
        {
                glPushMatrix();
                glTranslated(0.5,0.5*thickness,0.5);
                glScaled(1.0,thickness,1.0);
                glutSolidCube(1.0);
                glPopMatrix();
        }

        void tableLeg(double thick,double len)
        {
                glPushMatrix();
                glTranslated(0,len/2,0);
                glScaled(thick,len,thick);
                glutSolidCube(1.0);
                glPopMatrix();
        }

        void table(double topWid,double topThick,double
legThick,double legLen)
        {
                glPushMatrix();
                glTranslated(0,legLen,0);
                glScaled(topWid,topThick,topWid);
                glutSolidCube(1.0);
                glPopMatrix();
                double dist=0.95*topWid/2.0-legThick/2.0;
                glPushMatrix();
                glTranslated(dist,0,dist);
                tableLeg(legThick,legLen);
                glTranslated(0.0,0.0,-2*dist);
                tableLeg(legThick,legLen);
                glTranslated(-2*dist,0,2*dist);
                tableLeg(legThick,legLen);
                glTranslated(0,0,-2*dist);
                tableLeg(legThick,legLen);
                glPopMatrix();
```

```
        }

        void displaySolid(void)
        {
                GLfloat mat_ambient[]={0.7f,0.7f,0.7f,1.0f};
                GLfloat mat_diffuse[]={0.5f,0.5f,0.5f,1.0f};
                GLfloat mat_specular[]={1.0f,1.0f,1.0f,1.0f};
                GLfloat mat_shininess[]={50.0f};

                //The glMaterialfv function specifies material
parameters for the lighting model.
                glMaterialfv(GL_FRONT,GL_AMBIENT,mat_ambient);
                glMaterialfv(GL_FRONT,GL_DIFFUSE,mat_diffuse);
                glMaterialfv(GL_FRONT,GL_SPECULAR,mat_specular);
                glMaterialfv(GL_FRONT,GL_SHININESS,mat_shininess);
                GLfloat lightIntensity[]={0.7f,0.7f,0.7f,0.7f};
                GLfloat light_position[]={2.0f,6.0f,3.0f,0.0f};

                //The glLightfv function returns light source
parameter values.
                glLightfv(GL_LIGHT0,GL_POSITION,light_position);
                glLightfv(GL_LIGHT0,GL_DIFFUSE,lightIntensity);

                glMatrixMode(GL_PROJECTION);
                glLoadIdentity();
                double winHt=1.0;
                glOrtho(-winHt*64/48.0,winHt*64/48.0,-winHt,winHt,
0.1,100.0);
                glMatrixMode(GL_MODELVIEW);
                glLoadIdentity();
                gluLookAt(2.3,1.3,2.0,0.0,0.25,0.0,0.0,1.0,0.0);
                glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);

                glPushMatrix();
                glTranslated(0.4,0.4,0.6);
                glRotated(45,0,0,1);
                glScaled(0.08,0.08,0.08);
                glPopMatrix();
                glPushMatrix();
                glTranslated(0.6,0.38,0.5);
                glRotated(30,0,1,0);
                glutSolidTeapot(0.08);
                glPopMatrix();

                glPushMatrix();
                glTranslated(0.25,0.42,0.35);
                glPopMatrix();
                glPushMatrix();
                glTranslated(0.4,0,0.4);
                table(0.6,0.02,0.02,0.3);
                glPopMatrix();

                wall(0.02);
                glPushMatrix();
```

```
                glRotated(90.0,0.0,0.0,1.0);    //draw second wall
after rotating x axis by 90degre
                wall(0.02);
                glPopMatrix();
                glPushMatrix();
                glRotated(-90.0,1.0,0.0,0.0);   //draw floor
                wall(0.02);
                glPopMatrix();
                glFlush();
        }

        int main(int argc,char **argv)
        {
                glutInit(&argc,argv);
                glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB|
GLUT_DEPTH);
                glutInitWindowSize(640,480);
                glutInitWindowPosition(100,100);
                glutCreateWindow("Simple shaded scene consisting of
a teapot");
                glutDisplayFunc(displaySolid);
                glEnable(GL_LIGHTING);
                glEnable(GL_LIGHT0);
                glShadeModel(GL_SMOOTH);//Specifies a symbolic value
representing a shading technique. Accepted values are GL_FLAT and
GL_SMOOTH.
                glEnable(GL_DEPTH_TEST);
                glEnable(GL_NORMALIZE);
                glClearColor(0.1,0.1,0.1,0.0);
                glViewport(0,0,640,480);
                glutMainLoop();
        }

## Output:
*Commands for execution:-*

* Open a terminal and Change directory to the file location in both
the terminals.
* compile as gcc -lGLU -lGL -lglut teapot.c -o teapot
* If no errors, run as ./teapot

*Screenshots:-*

![Screenshot of Output](teapot.png)
```