Assignment - C1

Problem Statement - Write a C++/Java program to draw 3D cube and perform following on it using openGL
a) Scaling
b) Transfation
c) Rotation about one axis

Objective - ① To understand and implement open GL function
② To understand and learn the glut library.

Outcome - Learn and understand open GL for animating 3-D objects. Implement a simple cube with face of different colors using glut library.

H/w and S/w requirements - Core i3 processor, Fedora OS, Qt creator

Theory -

Open GL is a standard specification defining a cross language, cross platform API for writing application that produce 2D and 3D computer graphics. The interface consists of over 250 different function calls which can be used to draw complex 3D scenes from simple primitives. Open GL is a low level procedural API requiring the programmer

Most open GL commands either issue primitive to the graphics pipeline or configure how the pipeline processes these primitives.

Commands Evaluator Primitive Rasterisation Pretlagg
Assembly operation

Frame buffer

Display List	Pixel Operation	Texture memory

The command enter from left and proceed through a processing pipeline. Some commands specify geometric objects to the drawn and oth control how the objects are bounded during various processing stages.

Functions and features of open GL

1. Display list
2. Feedback
3. Alpha bending
4. Pixel operation
5. Texture mapping
6. color index mode
7. Polynomial evaluates
8. Scaling and rotation

Algorithm

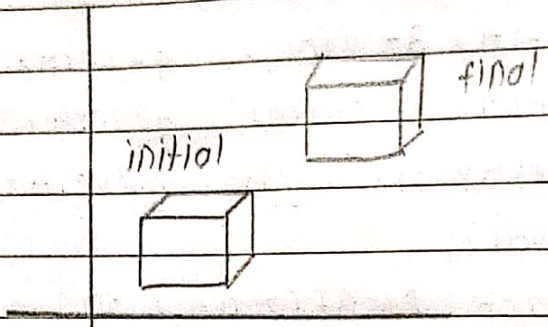
```
void DrawAxes() {
    glColor3f(1.0, 1.0, 1.0);
    glBegin(GL_Lines);
    glVertex3fv(org); glVertex3fv(XOP);
    glVertex3fv(org); glVertex3fv(YOP);
    glVertex3fv(org); glVertex3fv(ZOP);
    glEnd();
    glRasterPos3f(2.0, 0.0, 0.0);
    glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18 'X')
    glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18 'X')
    glRoasterPos3f(0.0, 0.0, 2);
    glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18 'X')

void display() {
    int v;
    glClear(GL_Color_Buffer_BIT);
    glLoadIdentity();
    glTranslate(0,0,-6); DrawAxes();
    glPushMatrix();
    glTranslate(d[0], d[1], d[2]);
    glScalef(x^5, y^5, z^5);
    glRotatef(xangle, 2, 0, 0);
    Draw_Box();
    glPopMatrix();
    glutSwapBuffer();
}
```

Test case

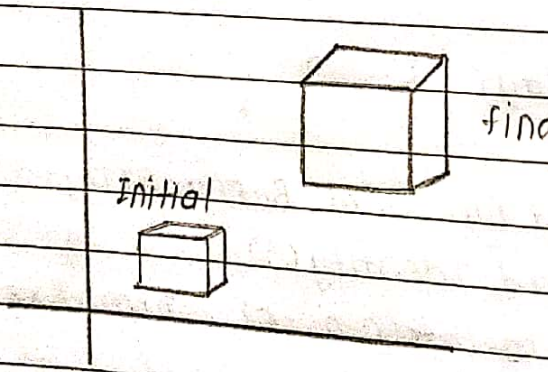
① Translate

Tx Ty Tz = 100, 100, 100



② Scaling

Sx Sy Sz = 3, 3, 3



Conclusion- Thus the 3D cube was implemented in open GL using glut library.