**EXAMPLE NO.1**

**#include "stdafx.h"**

**#include <GL/glut.h>**

**void init(void)**

**{**

**GLfloat mat\_specular[] = { 1.0, 1.0, 1.0, 1.0 };**

**GLfloat mat\_shininess[] = { 50.0 };**

**GLfloat light\_position[] = { 1.0, 1.0, 1.0, 0.0 };**

**glClearColor (0.0, 0.0, 0.0, 0.0);**

**glShadeModel (GL\_SMOOTH);**

**glMaterialfv(GL\_FRONT, GL\_SPECULAR, mat\_specular);**

**glMaterialfv(GL\_FRONT, GL\_SHININESS, mat\_shininess);**

**glLightfv(GL\_LIGHT0, GL\_POSITION, light\_position);**

**glEnable(GL\_LIGHTING);**

**glEnable(GL\_LIGHT0);**

**glEnable(GL\_DEPTH\_TEST);**

**}**

**void display(void)**

**{**

**glClear (GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);**

**glutSolidSphere (1.0, 20, 16);**

**glFlush ();**

**}**

**void reshape (int w, int h)**

**{**

**glViewport (0, 0, (GLsizei) w, (GLsizei) h);**

**glMatrixMode (GL\_PROJECTION);**

**glLoadIdentity();**

**if (w <= h)**

**glOrtho (-1.5, 1.5, -1.5\*(GLfloat)h/(GLfloat)w,**

**1.5\*(GLfloat)h/(GLfloat)w, -10.0, 10.0);**

**else**

**glOrtho (-1.5\*(GLfloat)w/(GLfloat)h,**

**1.5\*(GLfloat)w/(GLfloat)h, -1.5, 1.5, -10.0, 10.0);**

**glMatrixMode(GL\_MODELVIEW);**

**glLoadIdentity();**

**}**

**int main(int argc, char\*\* argv)**

**{**

**glutInit(&argc, argv);**

**glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB | GLUT\_DEPTH);**

**glutInitWindowSize (500, 500);**

**glutInitWindowPosition (100, 100);**

**glutCreateWindow (argv[0]);**

**init ();**

**glutDisplayFunc(display);**

**glutReshapeFunc(reshape);**

**glutMainLoop();**

**return 0;**

**}**

**EXAMPLE NO.2**

**#include "stdafx.h"**

**#include <GL/glut.h>**

**static int spin = 0;**

**void init(void)**

**{**

**glClearColor (0.0, 0.0, 0.0, 0.0);**

**glShadeModel (GL\_SMOOTH);**

**glEnable(GL\_LIGHTING);**

**glEnable(GL\_LIGHT0);**

**glEnable(GL\_DEPTH\_TEST);**

**}**

**/\* Here is where the light position is reset after the modeling**

**\* transformation (glRotated) is called. This places the**

**\* light at a new position in world coordinates. The cube**

**\* represents the position of the light.**

**\*/**

**void display(void)**

**{**

**GLfloat position[] = { 0.0, 0.0, 1.5, 1.0 };**

**glClear (GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);**

**glPushMatrix ();**

**glTranslatef (0.0, 0.0, -5.0);**

**glPushMatrix ();**

**glRotated ((GLdouble) spin, 1.0, 0.0, 0.0);**

**glLightfv (GL\_LIGHT0, GL\_POSITION, position);**

**glTranslated (0.0, 0.0, 1.5);**

**glDisable (GL\_LIGHTING);**

**glColor3f (0.0, 1.0, 1.0);**

**glutWireCube (0.1);**

**glEnable (GL\_LIGHTING);**

**glPopMatrix ();**

**glutSolidTorus (0.275, 0.85, 8, 15);**

**glPopMatrix ();**

**glFlush ();**

**}**

**void reshape (int w, int h)**

**{**

**glViewport (0, 0, (GLsizei) w, (GLsizei) h);**

**glMatrixMode (GL\_PROJECTION);**

**glLoadIdentity();**

**gluPerspective(40.0, (GLfloat) w/(GLfloat) h, 1.0, 20.0);**

**glMatrixMode(GL\_MODELVIEW);**

**glLoadIdentity();**

**}**

**void mouse(int button, int state, int x, int y)**

**{**

**switch (button) {**

**case GLUT\_LEFT\_BUTTON:**

**if (state == GLUT\_DOWN) {**

**spin = (spin + 30) % 360;**

**glutPostRedisplay();**

**}**

**break;**

**default:**

**break;**

**}**

**}**

**int main(int argc, char\*\* argv)**

**{**

**glutInit(&argc, argv);**

**glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB | GLUT\_DEPTH);**

**glutInitWindowSize (500, 500);**

**glutInitWindowPosition (100, 100);**

**glutCreateWindow (argv[0]);**

**init ();**

**glutDisplayFunc(display);**

**glutReshapeFunc(reshape);**

**glutMouseFunc(mouse);**

**glutMainLoop();**

**return 0;**

**}**

**EXAMPLE NO.3 :-**

**#include "stdafx.h"**

**#include <GL/glut.h>**

**GLfloat diffuseMaterial[4] = { 0.5, 0.5, 0.5, 1.0 };**

**void init(void)**

**{**

**GLfloat mat\_specular[] = { 1.0, 1.0, 1.0, 1.0 };**

**GLfloat light\_position[] = { 1.0, 1.0, 1.0, 0.0 };**

**glClearColor (0.0, 0.0, 0.0, 0.0);**

**glShadeModel (GL\_SMOOTH);**

**glEnable(GL\_DEPTH\_TEST);**

**glMaterialfv(GL\_FRONT, GL\_DIFFUSE, diffuseMaterial);**

**glMaterialfv(GL\_FRONT, GL\_SPECULAR, mat\_specular);**

**glMaterialf(GL\_FRONT, GL\_SHININESS, 25.0);**

**glLightfv(GL\_LIGHT0, GL\_POSITION, light\_position);**

**glEnable(GL\_LIGHTING);**

**glEnable(GL\_LIGHT0);**

**glColorMaterial(GL\_FRONT, GL\_DIFFUSE);**

**glEnable(GL\_COLOR\_MATERIAL);**

**}**

**void display(void)**

**{**

**glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);**

**glutSolidSphere(1.0, 20, 16);**

**glFlush ();**

**}**

**void reshape (int w, int h)**

**{**

**glViewport (0, 0, (GLsizei) w, (GLsizei) h);**

**glMatrixMode (GL\_PROJECTION);**

**glLoadIdentity();**

**if (w <= h)**

**glOrtho (-1.5, 1.5, -1.5\*(GLfloat)h/(GLfloat)w,**

**1.5\*(GLfloat)h/(GLfloat)w, -10.0, 10.0);**

**else**

**glOrtho (-1.5\*(GLfloat)w/(GLfloat)h,**

**1.5\*(GLfloat)w/(GLfloat)h, -1.5, 1.5, -10.0, 10.0);**

**glMatrixMode(GL\_MODELVIEW);**

**glLoadIdentity();**

**}**

**void mouse(int button, int state, int x, int y)**

**{**

**switch (button) {**

**case GLUT\_LEFT\_BUTTON:**

**if (state == GLUT\_DOWN) { /\* change red \*/**

**diffuseMaterial[0] += 0.1;**

**if (diffuseMaterial[0] > 1.0)**

**diffuseMaterial[0] = 0.0;**

**glColor4fv(diffuseMaterial);**

**glutPostRedisplay();**

**}**

**break;**

**case GLUT\_MIDDLE\_BUTTON:**

**if (state == GLUT\_DOWN) { /\* change green \*/**

**diffuseMaterial[1] += 0.1;**

**if (diffuseMaterial[1] > 1.0)**

**diffuseMaterial[1] = 0.0;**

**glColor4fv(diffuseMaterial);**

**glutPostRedisplay();**

**}**

**break;**

**case GLUT\_RIGHT\_BUTTON:**

**if (state == GLUT\_DOWN) { /\* change blue \*/**

**diffuseMaterial[2] += 0.1;**

**if (diffuseMaterial[2] > 1.0)**

**diffuseMaterial[2] = 0.0;**

**glColor4fv(diffuseMaterial);**

**glutPostRedisplay();**

**}**

**break;**

**default:**

**break;**

**}**

**}**

**int main(int argc, char\*\* argv)**

**{**

**glutInit(&argc, argv);**

**glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB | GLUT\_DEPTH);**

**glutInitWindowSize (500, 500);**

**glutInitWindowPosition (100, 100);**

**glutCreateWindow (argv[0]);**

**init ();**

**glutDisplayFunc(display);**

**glutReshapeFunc(reshape);**

**glutMouseFunc(mouse);**

**glutMainLoop();**

**return 0;**

**}**