```
#include<GL/glut.h>
#include<math.h>
#include<time.h>
#include<sys/timeb.h>
#define ESCAPE 27
int window;
float rtri = 0.0f;
//float rquad=0.0f;
void InitGL(int Width,int Height)
{
glClearColor(0.0f,0.0f,0.0f,0.0f);//set window color
//glClearDepth(1.0);
/*glDepthFunc(GL_LESS);
glEnable(GL_DEPTH_TEST);
glShadeModel(GL_SMOOTH);*/
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluPerspective(45.0f,(GLfloat)Width/(GLfloat)Height,0.1f,100.0f);
glMatrixMode(GL_MODELVIEW);
}
float ballX=-0.5f;
float ballY=0.0f;
float ballZ=0.0f;
void drawBall(void)
{
glColor3f(1.0,0.0,1.0);//set the ball color
glTranslatef(ballX,ballY,ballZ);
glRotatef(ballX,ballX,ballY,ballZ);
glutSolidSphere(0.3,50,50);
glTranslatef(ballX+1.5,ballY,ballZ);
glutSolidSphere(0.3,50,50);
```

```
}
void DrawGLScene()
{
//glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
glClear(GL_COLOR_BUFFER_BIT);
glLoadIdentity();
glTranslatef(rtri,0.0f,-6.0f);
glBegin(GL_POLYGON);
glColor3f(1.0f,0.0f,0.0f);//set triangle color
glVertex3f(-1.0f,1.0f,0.0);
glVertex3f(0.4f,1.0f,0.0f);
glVertex3f(1.0f,0.4f,0.0f);
//glColor3f(0.0f,1.0f,0.0f);
//glVertex3f(-1.0f,1.0f,0.0);
//glColor3f(0.4f,0.0f,1.0f);
//glVertex3f(1.0f,0.4f,0.0f);
glEnd();
drawBall();
rtri+=0.005f;
if(rtri>2)
{
rtri=-2.0f;
}
//rquad-=15.0f;
glutSwapBuffers();
}
void keyPressed(unsigned char key,int x,int y)
{
if(key==ESCAPE)
{
glutDestroyWindow(window);
exit(0);

}

}

int main(int argc,char **argv)

{

glutInit(&argc,argv);

//glutInitDisplayMode(GLUT_RGBA|GLUT_DOUBLE|GLUT_ALPHA|GLUT_DEPTH);

glutInitDisplayMode(GLUT_DOUBLE|GLUT_RGB);

glutInitWindowSize(640,480);

glutInitWindowPosition(0,0);

window=glutCreateWindow("Moving Object");

glutDisplayFunc(DrawGLScene);

glutIdleFunc(DrawGLScene);

glutKeyboardFunc(keyPressed);

InitGL(640,480);

glutMainLoop();

return(0);

}
```

OR

```
#include <iostream>
#include <math.h>
#include <time.h>
#include <GL/glut.h>

using namespace std;



int x=0;
int flag=0;
void init(){
   glClearColor(1.0,1.0,1.0,0.0);
   glMatrixMode(GL_PROJECTION);
   gluOrtho2D(0,640,0,480);
}

void object1(){

   glClear(GL_COLOR_BUFFER_BIT);

   glColor3f(1,0,0);
   glBegin(GL_POLYGON);
      glVertex2i(x,220);
      glVertex2i(x+40,220);
      glVertex2i(x+40,260);
      glVertex2i(x,260);
   glEnd();

   glutSwapBuffers();
```

```
}

void timer(int){

    glutPostRedisplay();
    glutTimerFunc(1000/60,timer,0);

    if(flag == 0){
        x = x+3;
    }
    if(flag == 1){
        x = x-3;
    }
    if(x==600){
        flag = 1;
    }
    if(x == 0){
        flag = 0;
    }

}
int main(int argc, char** argv){
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize(640,480);
    glutInitWindowPosition(200,200);
    glutCreateWindow("Animation");
    init();
    glutDisplayFunc(object1);
    glutTimerFunc(1000,timer,0);
    glutMainLoop();
    return 0;
}
```