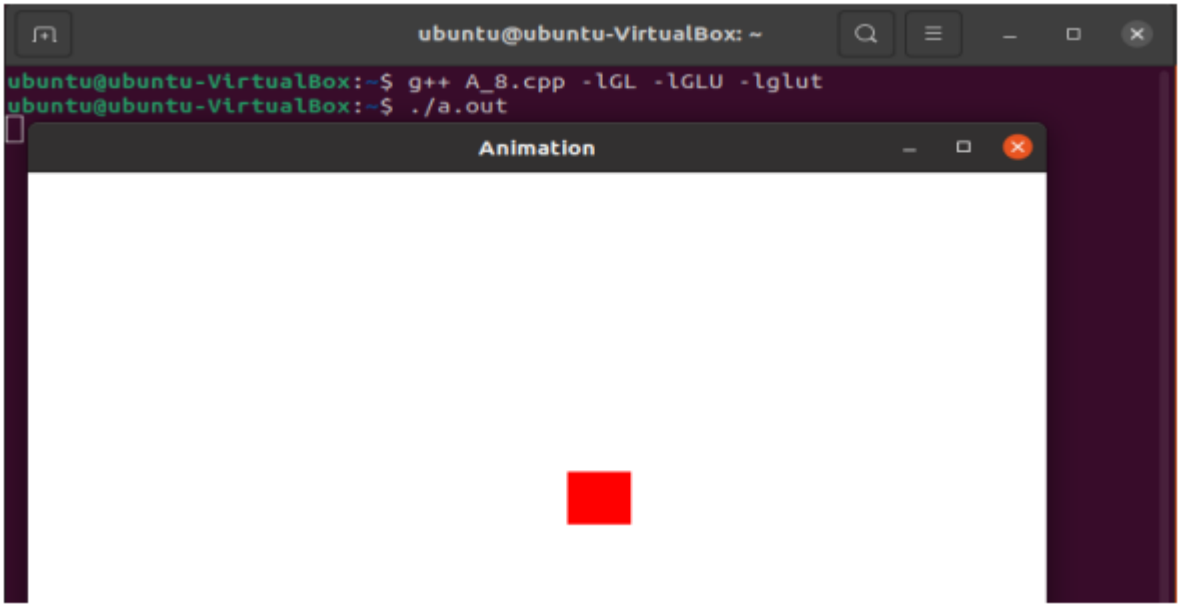
Experiment No 8

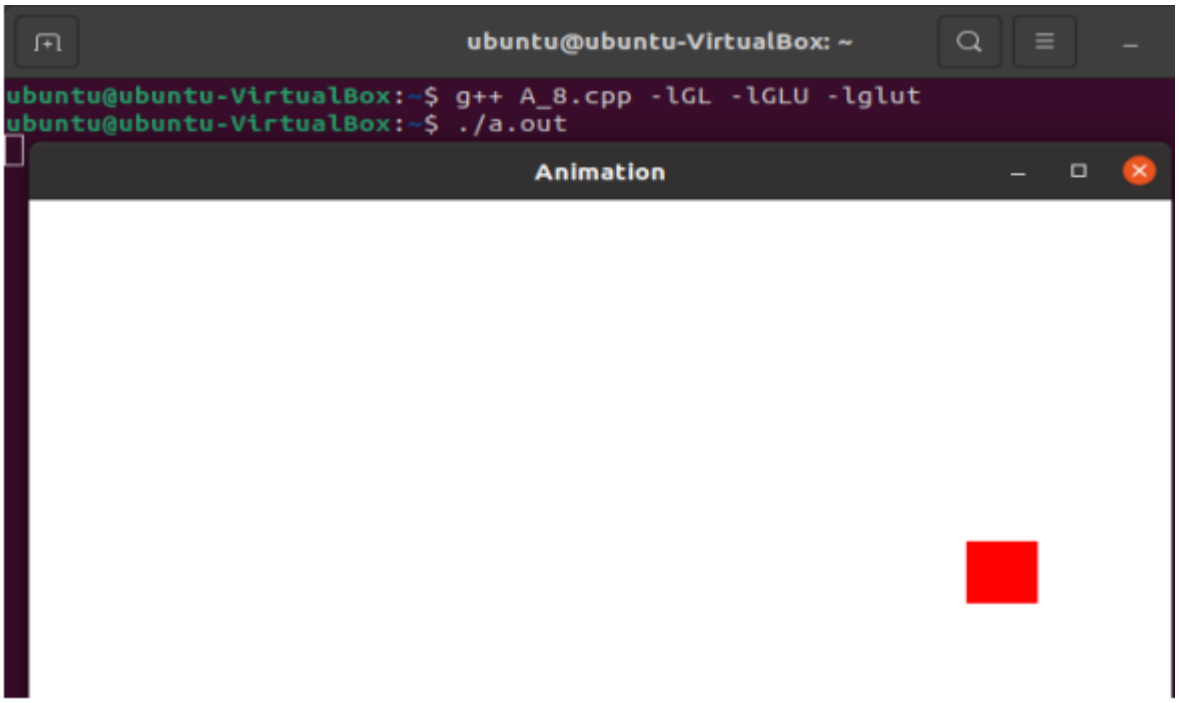
Title Implement animation principles for any object

```
#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;
}
```

OUTPUT


ubuntu@ubuntu-VirtualBox: ~
ubuntu@ubuntu-VirtualBox:~$ g++ A_8.cpp -lGL -lGLU -lglut
ubuntu@ubuntu-VirtualBox:~$ ./a.out
Animation


ubuntu@ubuntu-VirtualBox: ~
ubuntu@ubuntu-VirtualBox:~$ g++ A_8.cpp -lGL -lGLU -lglut
ubuntu@ubuntu-VirtualBox:~$ ./a.out
Animation