LAB - 8

8. Implement animation principles for any object.

Source Code :

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

float squareX = 0.0; // Initial x position of the square
void drawSquare() {
   glBegin(GL_QUADS);
   glColor3f(1.0, 0.0, 0.0); // Set color to red
   glVertex2f(squareX, 50);
   glVertex2f(squareX + 50, 50);
   glVertex2f(squareX + 50, 100);
   glVertex2f(squareX, 100);
   glEnd();
}

void display() {
   glClear(GL_COLOR_BUFFER_BIT);
   drawSquare();
   glFlush();
}

void update(int value) {
```

```
    squareX += 2.0;



    if (squareX > glutGet(GLUT_WINDOW_WIDTH))
        squareX = -50.0;



    glutPostRedisplay();



    glutTimerFunc(30, update, 0);
}

void init() {
    glClearColor(1.0, 1.0, 1.0, 1.0); // Set clear color to white
    gluOrtho2D(0, 400, 0, 200); // Set orthographic projection
}

int main(int argc, char **argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(400, 200);
    glutCreateWindow("Animation");
    init();
    glutDisplayFunc(display);
```

```
    glutTimerFunc(0, update, 0); // Start the animation timer
    glutMainLoop();
    return 0;
}
```

Output :