**PRACTICAL NO 10**

**Demonstrate the use of Anonymous Inner Class in Event Handling**

import javax.swing.\*;

import java.awt.event.\*;

import java.awt.\*;

class KeyEventExample extends JPanel {

private int x = 150; // Initial X position of the square

private int y = 150; // Initial Y position of the square

private JFrame frame;

KeyEventExample() {

frame = new JFrame("KeyEvent Example");

frame.setSize(400, 300);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// Add this panel to the frame

frame.add(this);

setFocusable(true); // Make the panel focusable to listen to key events

frame.setVisible(true);

addKeyListener(new KeyAdapter()

{

@Override

public void keyPressed(KeyEvent e) {

int keyCode = e.getKeyCode();

switch (keyCode) {

case KeyEvent.VK\_LEFT:

x -= 10; // Move left

break;

case KeyEvent.VK\_RIGHT:

x += 10; // Move right

break;

case KeyEvent.VK\_UP:

y -= 10; // Move up

break;

case KeyEvent.VK\_DOWN:

y += 10; // Move down

break;

}

// Repaint the panel to reflect the new position of the square

repaint();

}});

}

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

// Draw the square at the current (x, y) position

g.setColor(Color.BLUE);

g.fillRect(x, y, 80, 80);

}

public static void main(String[] args) {

new KeyEventExample();

}

}

**OUTPUT:**





