Experiment No. 12

Title: Write a program to demonstrate key and mouse event.

Objectives: 1 To learn key and mouse event
Theory:

Handling Keyboard Events-

A user interacts with the application by pressing keys either on the keyboard or by using mouse. A programmer should know which key the user has pressed on the keyboard or whether the mouse is moved, pressed, or released. These are also called '**events**'. Knowing these events will enable the programmer to write his code according to the key pressed or mouse event.
KeyListener interface of java. awt. event package helps to know which key is pressed or released by the user. It has 3 methods:
1]public void keyPressed (KeyEvent ke): This method is called when a key is pressed on the keyboard. This include any key on the keyboard along with special keys like function keys, shift, alter, caps lock, home, end, etc.
2] public void keyTyped (KeyEvent ke): This method is called when a key is typed on the keyboard. This is same as keyPressed () method but this method is called when general keys like A to Z or 1, to 9, etc. are typed. It cannot work with special keys. .
3] public void keyReleased (KeyEvent ke) : This method is called when a key is released,

KeyEvent class has the following methods to know which key is typed by the user:

1]char getKeyChar () : This method returns the key name (or character) related to the key pressed or released.
2]int getKeyCode () : This method returns an integer number which is the value of the key pressed by the user.
The following are the key codes for the keys on the keyboard. They are defined as constants in KeyEvent class. Remember VK represents Virtual Key.

o To represent keys from a to z: VK_ A to VK _Z
o To represent keys from 1 to 9: VK_ 0 to VK_9
o To represent keys from Fl to Fl2: VK_Fl to VK_F12
o To represent Home, End: VK_HOME, VK_END
o To represent PageUp, PageDown: VK_PAGE_UP, VK _PAGE _DOWN
o To represent Insert, Delete: VK_ INSERT, VK _DELETE
o To represent caps lock: VK_CAPS_LOCK
o To represent alter key: VK_ALT
o To represent Control key: VK_CONTROL
o To represent Shift key: VK_SHIFT
o To represent Tab key: VK_TAB
o To represent arrow keys: VK_LEFT, VK_RIGHT, VK_UP, VK_DOWN
o To represent Escape-key: VK_ESCAPE
o static String getKeyText(int keyCode)

This method returns a string describing the keyCode such as HOME, Fl, or A.

Handling Mouse Events-

The user may click, release, drag,. or move a mouse while interacting with the application. If the programmer knows what the user has done, he can write the code according to the mouse event. To trap the mouse events, MouseListener and MouseMotionListener interfaces of java.awt.event package are used.
MouseListener interface has the following methods:

O **void mouseClicked (MouseEvent e):** This method is invoked when the mouse button has been clicked (pressed and released) on a component..
O **void mouseEntered (MouseEvent e):** This method is invoked when the mouse enters a component.
O **void mouseExited (MouseEvent e):** This method is invoked when the mouse exits a component.
O void mousePressed (MouseEvent e): This method is invoked when a mouse button has been pressed on a component.
O **void mouseReleased (MouseEvent e):** This method is invoked when a mouse button has been released on a component.
MouseMotionListener interface has the following methods:

O **void mouseDragged (MouseEvent e):** This method is invoked when a mouse button is pressed on a component and then dragged.
O **void mouseMoved (MouseEvent e):** This method is invoked when the mouse cursor has been moved onto a component but no buttons have been pushed.

The MouseEvent class has the following methods:

O int getButton(): This method returns a value representing a mouse button, when it is clicked It returns 1 if left button is clicked, 2 if middle button, and 3 if right button is clicked;
O **int getClickCount ():** This method returns the number of mouse clicks associated with this event.
o int getX() : This method returns the horizontal x position of the event relative to the source component.
O int getY(): This method returns the vertical y position of the event relative to the source component.
Key concepts: KeyListener,keyEvent,MouseListener,MouseMotionListener

Algorithm for handling key Events:

1. Create class KeyBoardEvents that extends JFrame implemements KeyListener interface.
2. Define KeyBoardEvent class constructor to create contentPane & TextArea,
3.Add text area to content pane. Also add keyListener to text area.
4. Define keyPressed,keyReleased and keyTyped methods.
5. Create frame and define its properties.

Algorithm for handling mouse Events:

1. Create class MouseEvents that extends JFrame implemements MouseListener,MouseMotionListener interface.
2. Define MouseEvent class constructor to create contentPane & TextArea,
3.Add text area to content pane. Also add mouseListener and mouse motion Listener to text area.
4. Define mouseClicked, mouseEntered,mouseExited,mousePressed, mouseReleased,mouseDragged,mouseMoved and display method.
5. Create frame and define its properties.