CIS 36B – 17th Class / Lab Assignment – **10 Points-**

Student Name **Student ID** **Point Total**

Section 1: Define

Anonymous Class Listeners-

Anonymous classes enable you to make your code more concise. They enable you to declare and instantiate a class at the same time. They are like local classes except that they do not have a name. Use them if you need to use a local class only once.

MouseListener-

The listener interface for receiving "interesting" mouse events (press, release, click, enter, and exit) on a component. (To track mouse moves and mouse drags, use the `MouseMotionListener`.)

The class that is interested in processing a mouse event either implements this interface (and all the methods it contains) or extends the abstract `MouseAdapter` class (overriding only the methods of interest).

The listener object created from that class is then registered with a component using the component's `addMouseListener` method. A mouse event is generated when the mouse is pressed, released clicked (pressed and released). A mouse event is also generated when the mouse cursor enters or leaves a component. When a mouse event occurs, the relevant method in the listener object is invoked, and the `MouseEvent` is passed to it.

MouseMotionListener-

The listener interface for receiving mouse motion events on a component. (For clicks and other mouse events, use the `MouseListener`.)

The class that is interested in processing a mouse motion event either implements this interface (and all the methods it contains) or extends the abstract `MouseMotionAdapter` class (overriding only the methods of interest).

The listener object created from that class is then registered with a component using the component's `addMouseMotionListener` method. A mouse motion event is generated when the mouse is moved or dragged. (Many such events will be generated). When a mouse motion event occurs, the relevant method in the listener object is invoked, and the `MouseEvent` is passed to it.

KEY IDENTIFIERS	NAME KEY ON THE KEY BOARD
VK_HOME	Home
VK_END	End

CIS 36B – 17th Class / Lab Assignment – **10 Points-**

Student Name _____ **Student ID** _____ **Point Total**

VK_PGUP	PageUp
VK_PGDN	PageDown
VK_UP	Arrow key up
VK_DOWN	Arrow key down
VK_LEFT	Arrow Key Left
VK_RIGHT	Arrow key Right
VK_ESCAPE	Esc
VK_TAB	Tab
VK_CONTROL	Ctrl
VK_TAB	Tab
VK_CONTROL	Ctrl
VK_SHIFT	Shift
VK_BACK_SPACE	<-BackSpace
VK_CAPS_LOCK	Caps Lock
VK_NUM_LOCK	Num Lock
VK_ENTER	Enter
VK_UNDEFINED	KeyCode that's unKnown
VK_F1 to VK_F12	F1 to F12
VK_0 to VK_9	0 to 9
VK_A to VK_Z	A to Z

Task 1:

USE OBJECT ORIENTATED PROGRAM DESIGN TO SOLVE PROBLEM

(Display a Character) Write a program to get a character input from the key-board and display the character where the mouse points.

CIS 36B – 17th Class / Lab Assignment – **10 Points-**

Student Name _____ **Student ID** _____ **Point Total**

```
1  package mouse;
2
3  import java.awt.*;
4  import java.awt.event.MouseEvent;
5  import java.awt.event.MouseListener;
6  import java.awt.event.MouseMotionListener;
7  import java.awt.Graphics;
   import java.awt.Font;
   import static java.awt.font.TextAttribute.FONT;
10 import javax.swing.*;
11 import java.util.Scanner;
12
13 public class Mouse extends JFrame {
14
       private JPanel mousepanel;
       private JLabel statusbar;
17     private Graphics globalGraphics = null;
18     private String a;
19
20     @Override
       public void paint(Graphics g){
22         super.paint(g);
23         this.setSize(new Dimension(400,300));
24
25         if(globalGraphics == null){
26             globalGraphics = g.create();
27         }
28     }
29     public void drawDot(int x, int y){
30
31         globalGraphics.drawString(a, x, y);
32     }
33
34     public void init(){
35         Scanner input = new Scanner(System.in);
36         System.out.print("Please input a character: ");
37         this.a = input.next();
38     }
39
40     public Mouse(){
41         super("Window Title");
42
43         mousepanel = new JPanel();
44         mousepanel.setBackground(Color.WHITE);
45         add(mousepanel, BorderLayout.CENTER);
46
47         statusbar = new JLabel("Default");
48         add(statusbar, BorderLayout.SOUTH);
49
50
51         Handlerclass mousehandle = new Handlerclass();
52         mousepanel.addMouseListener(mousehandle);
53         statusbar.addMouseListener(mousehandle);
54     }
55
56     private class Handlerclass implements MouseListener, MouseMotionListener {
57
58         @Override
           public void mouseClicked(MouseEvent me) {
60             int mousex = me.getX();
```

CIS 36B – 17th Class / Lab Assignment – **10 Points-**

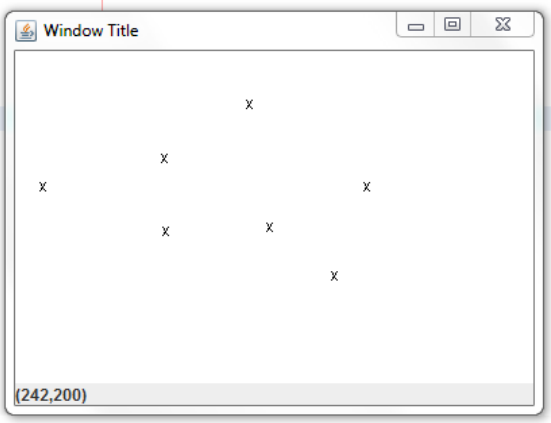
Student Name **Student ID** **Point Total**

```
61              int mousey = me.getY();
62              drawDot(mousex, mousey);
63
64              statusbar.setText("(" + String.valueOf(mousex) + ","  + String.valueOf(mousey) + ")");
65
66          }
67
68          @Override
            public void mousePressed(MouseEvent me) {
70              int mousex = me.getX();
71              int mousey = me.getY();
72              drawDot(mousex, mousey);
73
74              statusbar.setText("(" + String.valueOf(mousex) + ","  + String.valueOf(mousey) + ")");
75          }
76
77          @Override
            public void mouseReleased(MouseEvent me) {
79              //System.out.println("release");
80          }
81
82          @Override
            public void mouseEntered(MouseEvent me) {
84              //System.out.println("entered");
85          }
86
87          @Override
            public void mouseExited(MouseEvent me) {
89              //System.out.println("exited");
90          }
91
92          @Override
            public void mouseDragged(MouseEvent me) {
94              //System.out.println("dragged");
95          }
96
97          @Override
            public void mouseMoved(MouseEvent me) {
99              int mousex = me.getX();
100             int mousey = me.getY();
101
102             statusbar.setText("(" + String.valueOf(mousex) + ","  + String.valueOf(mousey) + ")");
103         }
104
105     }
106
107     public static void main(String[] args) {
108
109         Mouse sample = new Mouse();
110         sample.init();
111         sample.setPreferredSize(new Dimension(400, 300));
112         sample.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
113         sample.pack();
114         sample.setLocationRelativeTo(null);
115         sample.setFocusable(true);
116         sample.requestFocusInWindow();
117         sample.setVisible(true);
118
119     }
120
```

CIS 36B – 17th Class / Lab Assignment – **10 Points-**

Student Name _____ **Student ID** _____ **Point Total**

```
121     }
122
111             sample.setPreferredSize(new Dimension(400, 300));
112             sample.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
113             sample.pack();
114             sample.setLocationRelativeTo(null);
115             sample.setFocusable(true);
116             sample.requestFocusInWindow();
117             sample.setVisible(true);
118
119         }
120
121     }
122
```



Mouse › main

Output - mouse (run)

```
run:
Please input a character: x
```