OOP Theory

SE-101

JavaFX - Mouse and Keyboard Events

Submitted to:

Dr Madiha Liaqat

Submitted by:

Muhammad Abdul Rahim Attari

24-SE-27

Dated:

1. Description: This application demonstrates JavaFX event handling by responding to mouse and keyboard interactions. When users move the mouse or press keys, the application provides real-time feedback using a label.

2. Implemented Features:

- Mouse entered/exited stage → updates label and background color.
- Mouse click → shows (x, y) and counts total clicks.
- Keyboard key press → identifies key type and counts total presses.

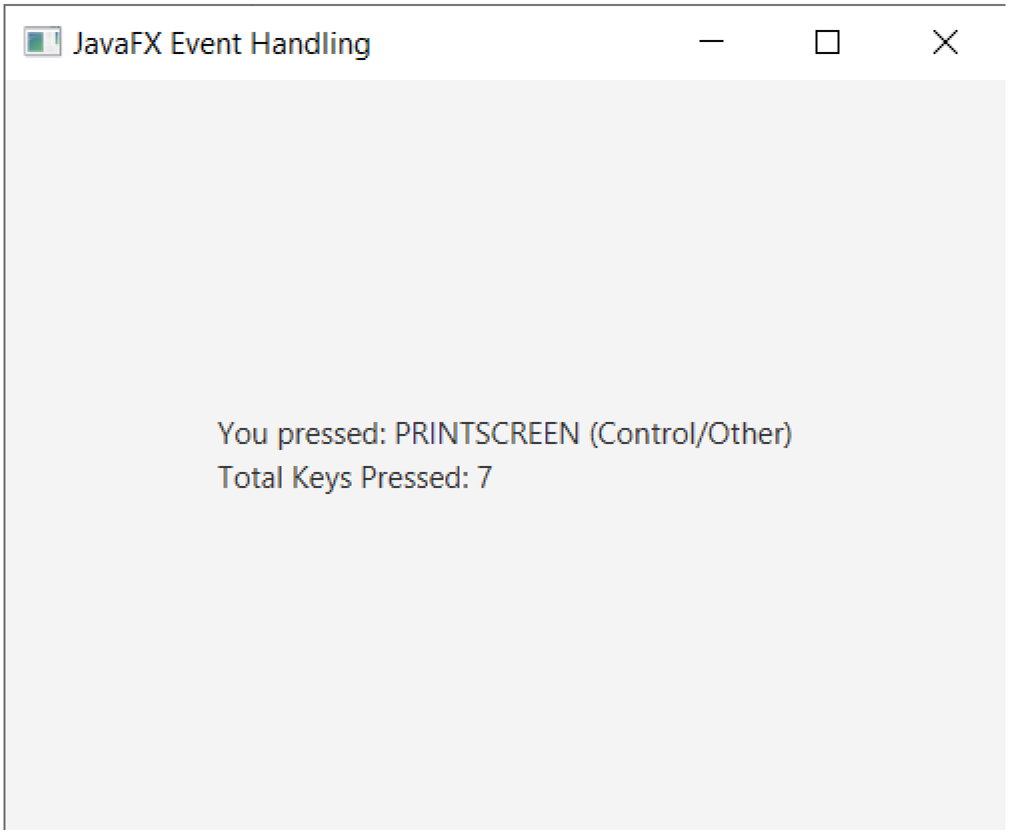
3. Tools Used:

- JavaFX
- StackPane for layout
- Label for output messages
- EventHandlers for input

Code:

```
// Name : Muhammad Abdul Rahim Attari
// Registration No: 24-SE-27
// Assignment: JavaFX - Mouse and Keyboard Events

package com.example.javafxlab12;
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.StackPane;
import javafx.scene.paint.Color;
import javafx.stage.Stage;

public class OOPAssignment extends Application {

    Label infoLabel = new Label("Interact with the window!");
    int mouseClickCount = 0;
    int keyPressCount = 0;

    @Override
    public void start(Stage primaryStage) {
        StackPane root = new StackPane();
        root.getChildren().add(infoLabel);

        Scene scene = new Scene(root, 400, 300);

        // Mouse Entered
        scene.setOnMouseEntered((MouseEvent e) -> {
            infoLabel.setText("Mouse entered the stage");
            scene.setFill(Color.LIGHTGREEN); // Bonus: change background
        });

        // Mouse Exited
        scene.setOnMouseExited((MouseEvent e) -> {
            infoLabel.setText("Mouse exited the stage");
            scene.setFill(Color.LIGHTCORAL); // Bonus: change background
        });

        // Mouse Clicked
        scene.setOnMouseClicked((MouseEvent e) -> {
            mouseClickCount++;
            infoLabel.setText("Mouse clicked at: X=" + e.getSceneX() + ", Y=" +
e.getSceneY() +
                    "\nTotal Clicks: " + mouseClickCount); // Bonus: count clicks
        });

        // Key Pressed
        scene.setOnKeyPressed((KeyEvent e) -> {
            keyPressCount++;
            String keyText = e.getText();
            String type;

            if (keyText.matches("[a-zA-Z]")) {
                type = "Letter";
            } else if (keyText.matches("[0-9]")) {
                type = "Number";
            } else {
                type = "Control/Other";
            }

            infoLabel.setText("You pressed: " + e.getCode() + " (" + type + ")" +
                    "\nTotal Keys Pressed: " + keyPressCount); // Bonus: count keys
        });

        primaryStage.setTitle("JavaFX Event Handling");
        primaryStage.setScene(scene);
        primaryStage.show();

        // Important: Request focus so keyboard events work
        root.requestFocus();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

Screen Shot:



```
You pressed: PRINTSCREEN (Control/Other)
Total Keys Pressed: 7
```

Note:

The program is handling the events, so when I click PrtScr (Print Screen), it's shown in the output in the scene, and other mouse events will not be captured.