Southern Luzon State University
College of Engineering
Computer Engineering Department

CPE13 Object Oriented Programming

Activity 8: Java Event and Action Listener

Name: REIMARC G. CORPUZ Date: NOV. 15, 2022

Section: BSCPE 3GF Score:_______________

1.1 Introduction

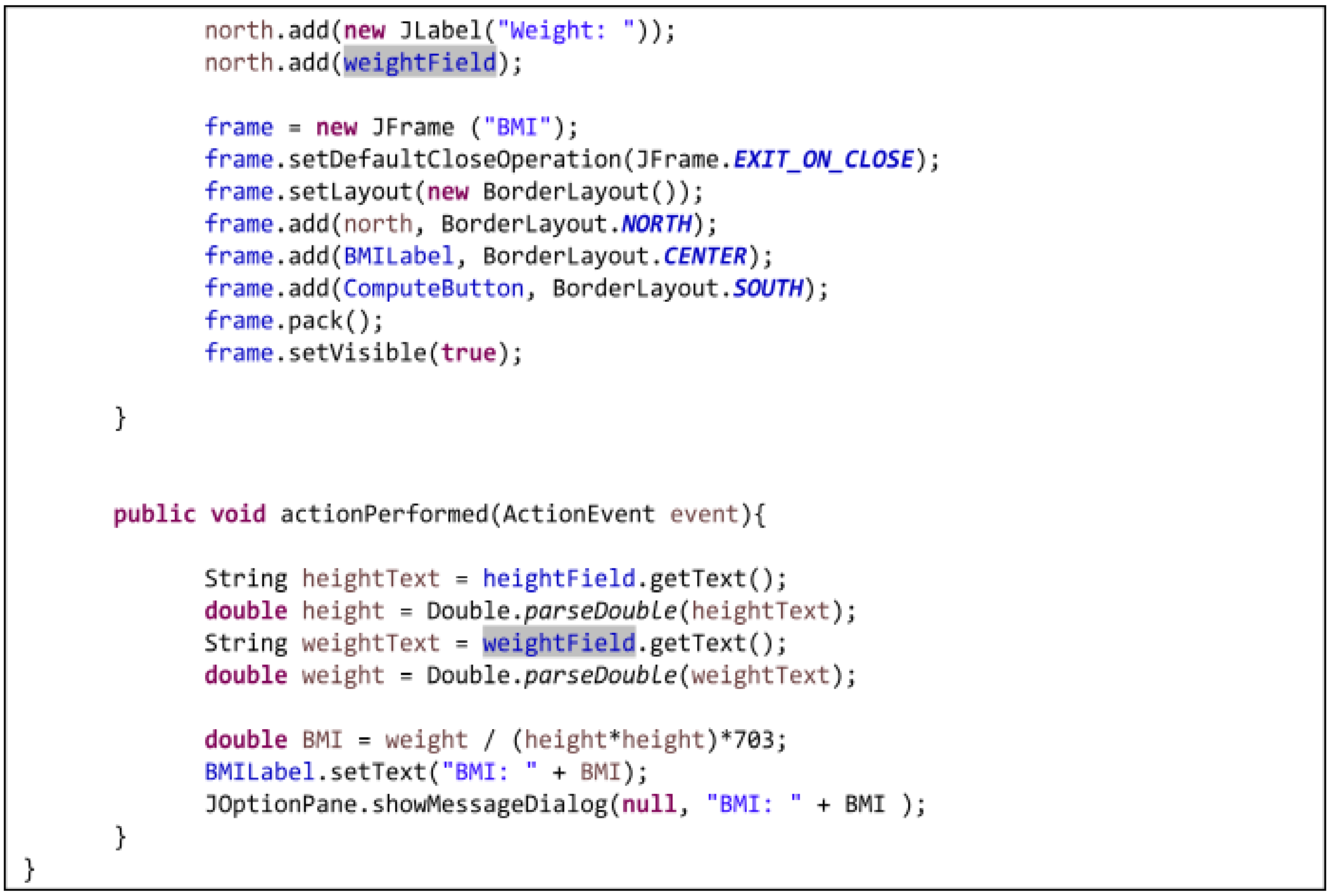
Aside from Layout Managers, Panels, Frames and its Components, Java also handles different events and actions to be performed whenever a user interact with the GUI you created. You will need another import packages, interfaces and methods to do this. The import package which is "import java.awt.event.*;" is different from "import java.awt.*;" so you will need to import both packages. You can also utilize this event using implementation of ActionListener and a class to perform the event. The usual coding have a format of creating the viewable part or the physical part before the action.

1.2 Objective

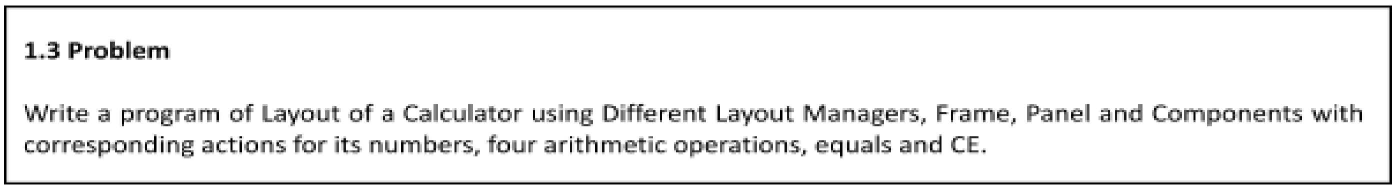
- To use Java programming language to create a program that uses Action Listener.
- To conceptualize the process and manipulate the program
- To distinguish different parts of GUI Creation particularly the creation of layout managers, panels, frames and its components with its corresponding actions and events.

Sample Program:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class GUIActivity51 implements ActionListener{

	public static void main(String[] args){

		GUIActivity51 GUI = new GUIActivity51();

	}

	private JFrame frame;
	private JTextField heightField,weightField;
	private JLabel BMILabel;
	private JButton ComputeButton;

	public GUIActivity51(){

		heightField = new JTextField(5);
		weightField = new JTextField(5);
		BMILabel = new JLabel("Type your height and weight");
		ComputeButton = new JButton("Compute");
		ComputeButton.addActionListener(this);


		JPanel north = new JPanel (new GridLayout(2,2));
		north.add(new JLabel("Height: "));
		north.add(heightField);
```



```
        north.add(new JLabel("Weight: "));
        north.add(weightField);

        frame = new JFrame ("BMI");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new BorderLayout());
        frame.add(north, BorderLayout.NORTH);
        frame.add(BMILabel, BorderLayout.CENTER);
        frame.add(ComputeButton, BorderLayout.SOUTH);
        frame.pack();
        frame.setVisible(true);

    }


    public void actionPerformed(ActionEvent event){

        String heightText = heightField.getText();
        double height = Double.parseDouble(heightText);
        String weightText = weightField.getText();
        double weight = Double.parseDouble(weightText);

        double BMI = weight / (height*height)*703;
        BMILabel.setText("BMI: " + BMI);
        JOptionPane.showMessageDialog(null, "BMI: " + BMI );
    }
}
```



1.3 Problem

Write a program of Layout of a Calculator using Different Layout Managers, Frame, Panel and Components with corresponding actions for its numbers, four arithmetic operations, equals and CE.

1.4 Questions

1. **Is JOptionPane important in events and actions? Why?**

Yes. JOptionPane makes it simple to display a typical dialog box that asks users for a value or provides them with information.

2. **What is the purpose of implementing Actionlistener to the class?**

To specify what should be done when a user performs a specific action, you construct an action listener. Every time a user does an action, an action event takes place. Examples include when a user clicks a button, selects an item from a menu, or enters text. All action listeners registered on the pertinent component receive an actionPerformed message as a result.

3. **What other events can you enumerate? Explain each.**

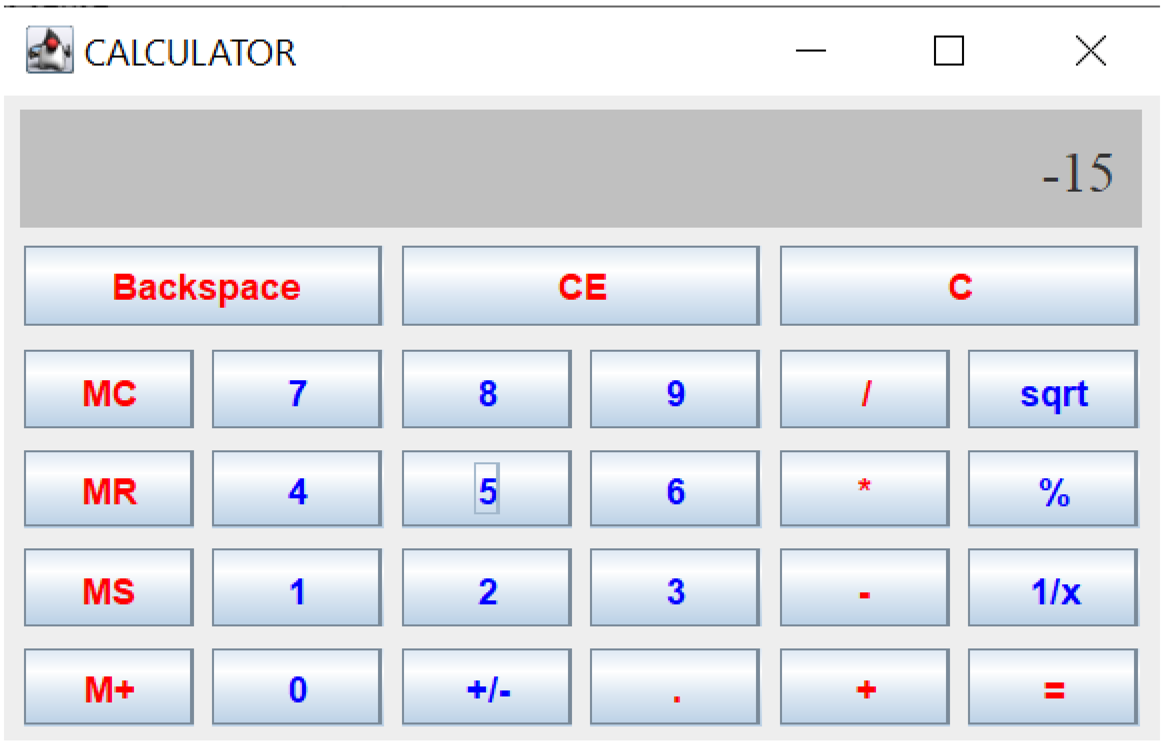
1. AdjustmentEvent (adjustmentValueChanged()) - occurs by an adjustable object like a scrollbar.
2. ComponentEvent (componentResized(), componentMoved(), componentShown() and componentHidden()) - An event occurs when a component moved, changed its visibility or the size changed.
3. ContainerEvent (componentRemoved() and componentAdded()) - The event is fired when a component is added or removed from a container.
4. FocusEvent (focusLost() and focusGained()) - Focus events include focus, focusout, focusin, and blur.
5. ItemEvent (itemStateChanged()) - Item event occurs when an item is selected.

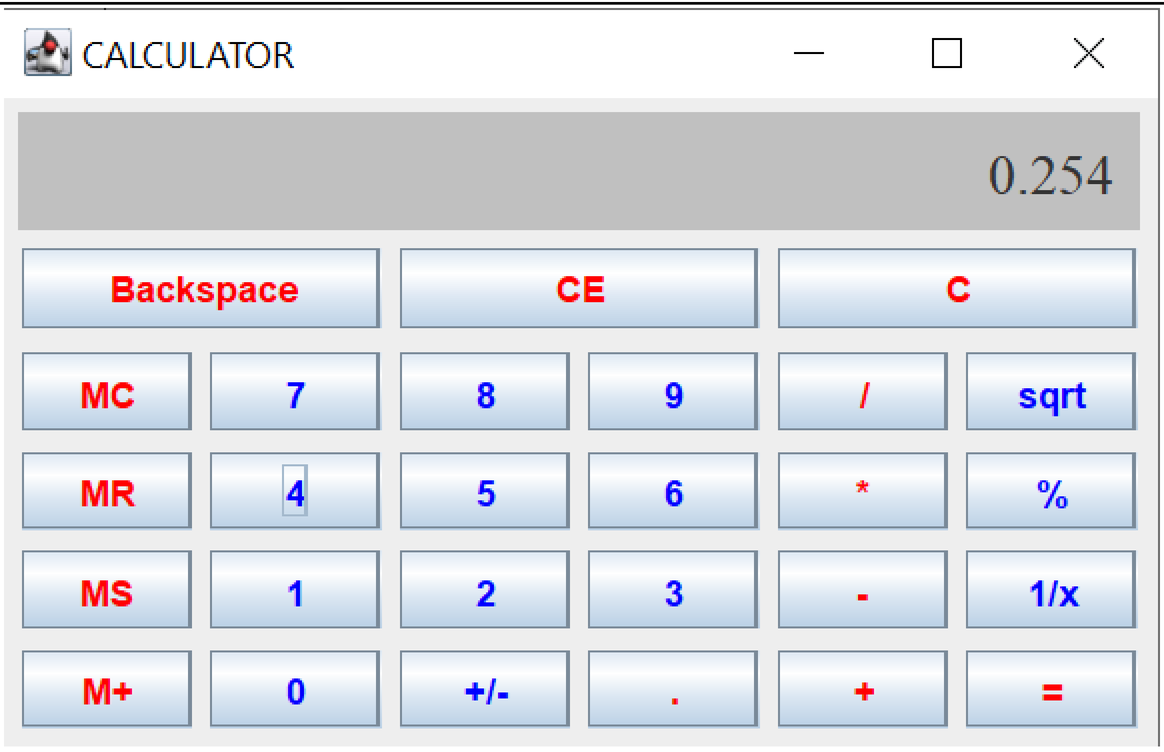
6. KeyEvent (keyPressed(), keyReleased(), and keyTyped()) - A key event occurs when the user presses a key on the keyboard.
7. MouseEvent (mouseClicked(), mousePressed(), mouseEntered(), mouseExited() and mouseReleased() are the mouseListener methods. mouseDregged() and mouseMoved() are the MouseMotionListener() methods) - A mouse event occurs when the user interacts with the mouse.
8. MouseWheelEvent (mouseWheelMoved()) - occurs when the mouse wheel rotates in a component.
9. TextEvent (textChanged()) - occurs when an object's text changes.
10. WindowEvent (windowActivated(), windowDeactivated(), windowOpened(), windowClosed(), windowClosing(), windowIconfied() and windowDeiconified()) - Window events occur when a window's status is changed.

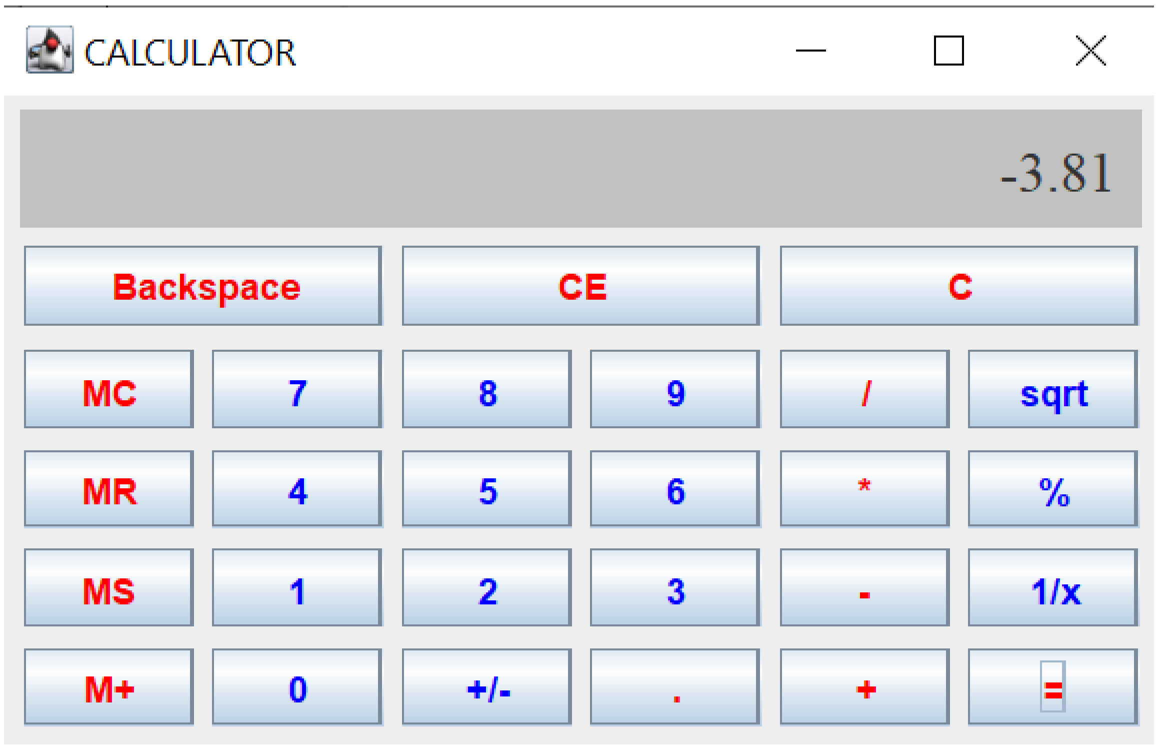
1.5 Conclusion

After the activity, I concluded/understood that one of the most essential ideas in Java is an event. A Java event is a state change brought about by actions that change an object's or behavior's state. The relevant event object is created when a user performs an action, such as clicking a button, selecting an option from a combo box, or entering text into a text field. The object that "listens" for events and handles them as they occur is an event listener. It should be noted that several listeners and event sources can communicate with one another. For instance, if they are of the same kind, a single listener can register numerous events. This implies that one event listener may manage all the events for a similar set of components that carry out the same kind of operation. Similar to that, if it makes sense for the program's architecture, a single event can be connected to numerous listeners.

Output: Screenshot only (2 screenshots 1-operation and 2-output)







Note: If I select the first number the input number will display in the text field, and then if I select the operator the first input number will be stored in the memory and will not be visible. The second input number will then display to the text field. The answer will display in the text field once I select the equal button. So, I did three screenshots of display output.

The sample display output solve: -15 * 0.254 = -3.81

**See the attached file for the sample simulation of display output.*

Note: Upload your PDF and JAVA file.

Write a concept/idea for the application development.

1. **Concept must be related with CPE.**
2. **Concept should have the following application genre:**
    a. **Group 1: Arcade Game**
    b. **Group 2: Quiz type (Word Game)**
    c. **Group 3: Board Game**
    d. **Group 4: Picture-based Game**
3. **Application Title and Description.**