	PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE -43		
	Department of Electronics and Telecommunication Engineering		
	ASSESSMENT YEAR: - 2021-22		**CLASS: - TE-V**
	Subject: - Advanced Java Programming		
Expt. No: 04	**LAB Ref: ETC/202122/**	**ROLL NO: 32147**	**SUBMISION DATE:**

Title: - Arithmetic Calculator

Write a program in Java to design GUI for calculator to perform arithmetic operators

Objectives: -

1. To learn the concepts of action listener and integration of multiple classes

Theory

1. Frame:

The Frame is the container that contain title bar and border and can have menu bars.

It can have other components like button, text field, scrollbar etc. Frame is most widely used container while developing an AWT application.

A Frame is a top-level window with a title and a border.

The size of the frame includes any area designated for the border. The dimensions of the border area may be obtained using the getInsets method, however, since these dimensions are platform-dependent, a valid insets value cannot be obtained until the frame is made displayable by either calling pack or show A frame may have its native decorations (i.e. Frame and Titlebar) turned off with setUndecorated. This can only be done while the frame is not displayable.

```
Frame f = new Frame(GraphicsConfiguration gc);
    Rectangle bounds = gc.getBounds();
    f.setLocation(10 + bounds.x, 10 + bounds.y);
```

Frame is a subclass of Window and contains title, border and menu bars.

It comes with a resizing canvas and is the most widely used container for developing AWT applications.

It is capable of holding various components such as buttons, text fields, scrollbars, etc.

2. TextField:

The textField component allows the user to edit single line of text.When the user types a key in the text field the event is sent to the TextField.

The key event may be key pressed, Key released or key typed.

The key event is passed to the registered KeyListener.

t is also possible to for an ActionEvent if the ActionEvent is enabled on the textfield then ActionEvent may be fired by pressing the return key.

public class TextField
extends TextComponent

S.N.	Constructor & Description
1	**TextField()** Constructs a new text field.
2	**TextField(int columns)** Constructs a new empty text field with the specified number of columns.
3	**TextField(String text)** Constructs a new text field initialized with the specified text.
4	**TextField(String text, int columns)** Constructs a new text field initialized with the specified text to be displayed, and wide enough to hold the specified number of columns.

3. Button:

A button is basically a control component with a label that generates an event when pushed

The **Button** class is used to create a labeled button that has platform independent implementation. The application result in some action when the button is pushed.

When we press a button and release it, AWT sends an instance of **ActionEvent** to that button by calling **processEvent** on the button.

Following table shows the types of Button class constructors

Sr. no.	Constructor	Description
1.	Button( )	It constructs a new button with an empty string i.e. it has no label.
2.	Button (String text)	It constructs a new button with given string as its label.

4. ActionListener:

The Java ActionListener is notified whenever you click on the button or menu item. It is notified against ActionEvent. The ActionListener interface is found in java.awt.event package. It has only one method: actionPerformed().

The class which processes the ActionEvent should implement this interface.The object of that class must be registered with a component.

When the action event occurs, that object's actionPerformed method is invoked.

Following is the declaration for **java.awt.event.ActionListener** interface:

```
public interface ActionListener
  extends EventListener
```

S.N.	Method & Description
1	**void actionPerformed(ActionEvent e)** Invoked when an action occurs.

5. Event class:

The Event classes represent the event. Java provides us various Event classes but we will discuss those which are more frequently used.

It is the root class from which all event state objects shall be derived.

All Events are constructed with a reference to the object, the **source**, that is logically deemed to be the object upon which the Event in question initially occurred upon.This class is defined in java.util package.

Following is the declaration for **java.util.EventObject** class:

```
public class EventObject
  extends Object
    implements Serializable
```

Class methods:

S.N.	Method & Description
1	**Object getSource()** The object on which the Event initially occurred.
2	**String toString()** Returns a String representation of this EventObject.

1. AWTEvent

It is the root event class for all AWT events. This class and its subclasses supercede the original java.awt.Event class.

2. ActionEvent

The ActionEvent is generated when button is clicked or the item of a list is double clicked.

3. InputEvent

The InputEvent class is root event class for all component-level input events.

4. KeyEvent

On entering the character the Key event is generated.

5. MouseEvent

This event indicates a mouse action occurred in a component.

6. TextEvent

The object of this class represents the text events.

7. WindowEvent

The object of this class represents the change in state of a window

8. MouseMotionEvent

The object of this class represents the change in state of a window.

AWT is one of the way to do GUI programming in java. It is a package that contains all classes and interface that are needed to implement GUI

ArithOperators in Java:

The Java programming language supports various arithmetic operators for all floating-point and integer numbers. These operators are + (addition), - (subtraction), * (multiplication), / (division), and % (modulo).

Label:

The object of the Label class is a component for placing text in a container

It is used to display a single line of **read only text**. The text can be changed by a programmer but a user cannot edit it directly.

It is called a passive control as it does not create any event when it is accessed. To create a label, we need to create the object of **Label** class.

The java.awt.Component class has following fields:

1. **static int LEFT:** It specifies that the label should be left justified.
2. **static int RIGHT:** It specifies that the label should be right justified.
3. **static int CENTER:** It specifies that the label should be placed in center.

Sr. no.	Constructor	Description
1.	Label()	It constructs an empty label.
2.	Label(String text)	It constructs a label with the given string (left justified by default).
3.	Label(String text, int alignement)	It constructs a label with the specified string and the specified alignment.

Diagram:

Learning Outcomes: -		
	1	To understand the event classes
	2	To learn the concept of ActionListener interface
	3	To use the method ActionPerformed
	4	To understand the integration of multiple classes
	5	To use the arithmetic operators

Continuous Assessment

RPP (out of 5)	SPO (out of 5)	Total (Out of 10)	Sign
			Date: -

#(RPP – Regularity, Punctuality, Performance), (SPO – Submission, Presentation, Oral)

Important Questions: -

1. **What is interface? Explain ActionListener interface.**
2. **Explain the ActionPerformed method and give one example.**
3. **List down event classses**
4. **How to integrate multiple classes in java?**
5. **Can we implement ActionPerfomed method using adapter class?**