4. LABEL:

The object of 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 an application but a user cannot edit it directly.

Example:

EVENT HANDLING

Change in the state of an object is known as event i.e. event describes the change in state of source. Events are generated as result of user interaction with the graphical user interface components. For example, clicking on a button, moving the mouse, , selecting entering a character through a keyboard an item from a list, scrolling the page are the activities that cause an event to happen.

The events can be broadly classified into two categories:

- Foreground Events - Those events which require the direct interaction of the user.They are generated as consequences of a person interacting with the graphical components in Graphical User Interface. For example, clicking on a button, moving the mouse, entering a character through the keyboard,selecting an item from a list, scrolling the page etc.
- Background Events - Those events that require the interaction of the end user are known as background events. Operating system interrupts, hardware or software failure, timer expires, and operation completion are the examples of background events.

Event Handling is the mechanism that controls the event and decides what should happen if an event occurs. Java Uses the Delegation Event Model to handle the events. This model defines the standard mechanism to generate and handle the events.Let's have a brief introduction to this model.

The Delegation Event Model has the following key participants namely:

- Source - The source is an object on which event occurs. Source is responsible for providing information of the occurred event to it's handler. Java provides classes for source object.
- Listener - It is also known as event handler.Listener is responsible for generating response to an event. From a Java implementation point of view the listener is also an object. Listener waits until it receives an event. Once the event is received , the listener processes the event and then returns.

The java.awt.event package provides many event classes and Listener interfaces for event handling.

1. Event Classes Listener Interfaces
2. ActionEvent ActionListener
3. MouseEvent MouseListener and MouseMotionListener
4. MouseWheelEvent MouseWheelListener
5. KeyEvent KeyListener
6. ItemEvent ItemListener
7. TextEvent TextListener
8. AdjustmentEvent AdjustmentListener
9. WindowEvent WindowListener
10. ComponentEvent ComponentListener
11. ContainerEvent ContainerListener
12. FocusEvent FocusListener

Steps to perform Event Handling

Following steps are required to perform event handling:

1. Register the component with the Listener