**[next →](https://www.javatpoint.com/event-handling-in-java)**

Java AWT Tutorial

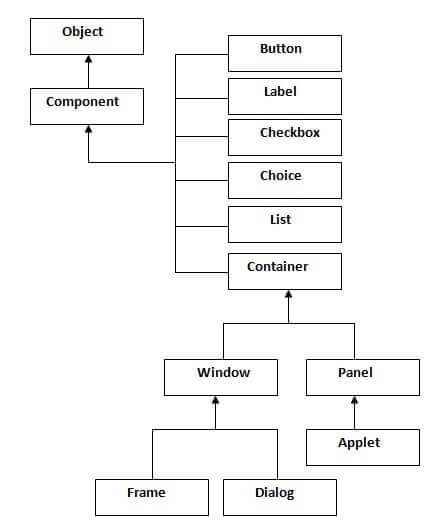
**Java AWT** (Abstract Window Toolkit) is *an API to develop GUI or window-based applications* in java.

Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system. AWT is heavyweight i.e. its components are using the resources of OS.

The java.awt [package](https://www.javatpoint.com/package) provides [classes](https://www.javatpoint.com/object-and-class-in-java) for AWT api such as [TextField](https://www.javatpoint.com/java-awt-textfield), [Label](https://www.javatpoint.com/java-awt-label), [TextArea](https://www.javatpoint.com/java-awt-textarea), RadioButton, [CheckBox](https://www.javatpoint.com/java-awt-checkbox), [Choice](https://www.javatpoint.com/java-awt-choice), [List](https://www.javatpoint.com/java-awt-list) etc.

### Java AWT Hierarchy

The hierarchy of Java AWT classes are given below.



### Container

The Container is a component in AWT that can contain another components like [buttons](https://www.javatpoint.com/java-awt-button), textfields, labels etc. The classes that extends Container class are known as container such as Frame, Dialog and Panel.

### Window

The window is the container that have no borders and menu bars. You must use frame, dialog or another window for creating a window.

### Panel

The Panel is the container that doesn't contain title bar and menu bars. It can have other components like button, textfield etc.

### Frame

The Frame is the container that contain title bar and can have menu bars. It can have other components like button, textfield etc.

### Useful Methods of Component class

|  |  |
| --- | --- |
| **Method** | **Description** |
| public void add(Component c) | inserts a component on this component. |
| public void setSize(int width,int height) | sets the size (width and height) of the component. |
| public void setLayout(LayoutManager m) | defines the layout manager for the component. |
| public void setVisible(boolean status) | changes the visibility of the component, by default false. |

### Java AWT Example

To create simple awt example, you need a frame. There are two ways to create a frame in AWT.

* By extending Frame class (inheritance)
* By creating the object of Frame class (association)

## AWT Example by Inheritance

Let's see a simple example of AWT where we are inheriting Frame class. Here, we are showing Button component on the Frame.

1. **import** java.awt.\*;
2. **class** First **extends** Frame{
3. First(){
4. Button b=**new** Button("click me");
5. b.setBounds(30,100,80,30);// setting button position
6. add(b);//adding button into frame
7. setSize(300,300);//frame size 300 width and 300 height
8. setLayout(**null**);//no layout manager
9. setVisible(**true**);//now frame will be visible, by default not visible
10. }
11. **public** **static** **void** main(String args[]){
12. First f=**new** First();
13. }}

[download this example](https://static.javatpoint.com/src/awt/first.zip)

The setBounds(int xaxis, int yaxis, int width, int height) method is used in the above example that sets the position of the awt button.



## AWT Example by Association

Let's see a simple example of AWT where we are creating instance of Frame class. Here, we are showing Button component on the Frame.

1. **import** java.awt.\*;
2. **class** First2{
3. First2(){
4. Frame f=**new** Frame();
5. Button b=**new** Button("click me");
6. b.setBounds(30,50,80,30);
7. f.add(b);
8. f.setSize(300,300);
9. f.setLayout(**null**);
10. f.setVisible(**true**);
11. }
12. **public** **static** **void** main(String args[]){
13. First2 f=**new** First2();
14. }}

[download this example](https://static.javatpoint.com/src/awt/first2.zip)



Next Topic[Event Handling in java](https://www.javatpoint.com/event-handling-in-java)

# Event and Listener (Java Event Handling)

|  |
| --- |
| Changing the state of an object is known as an event. For example, click on button, dragging mouse etc. The java.awt.event package provides many event classes and Listener interfaces for event handling. |

## Java Event classes and Listener interfaces

|  |  |
| --- | --- |
| **Event Classes** | **Listener Interfaces** |
| ActionEvent | ActionListener |
| MouseEvent | MouseListener and MouseMotionListener |
| MouseWheelEvent | MouseWheelListener |
| KeyEvent | KeyListener |
| ItemEvent | ItemListener |
| TextEvent | TextListener |
| AdjustmentEvent | AdjustmentListener |
| WindowEvent | WindowListener |
| ComponentEvent | ComponentListener |
| ContainerEvent | ContainerListener |
| FocusEvent | FocusListener |

## Steps to perform Event Handling

Following steps are required to perform event handling:

1. Register the component with the Listener

## Registration Methods

For registering the component with the Listener, many classes provide the registration methods. For example:

* **Button**
  + public void addActionListener(ActionListener a){}
* **MenuItem**
  + public void addActionListener(ActionListener a){}
* **TextField**
  + public void addActionListener(ActionListener a){}
  + public void addTextListener(TextListener a){}
* **TextArea**
  + public void addTextListener(TextListener a){}
* **Checkbox**
  + public void addItemListener(ItemListener a){}
* **Choice**
  + public void addItemListener(ItemListener a){}
* **List**
  + public void addActionListener(ActionListener a){}
  + public void addItemListener(ItemListener a){}

### Java Event Handling Code

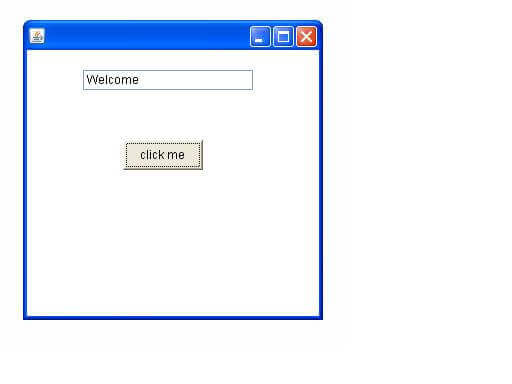
We can put the event handling code into one of the following places:

1. Within class
2. Other class
3. Anonymous class

### Java event handling by implementing ActionListener

1. **import** java.awt.\*;
2. **import** java.awt.event.\*;
3. **class** AEvent **extends** Frame **implements** ActionListener{
4. TextField tf;
5. AEvent(){
7. //create components
8. tf=**new** TextField();
9. tf.setBounds(60,50,170,20);
10. Button b=**new** Button("click me");
11. b.setBounds(100,120,80,30);
13. //register listener
14. b.addActionListener(**this**);//passing current instance
16. //add components and set size, layout and visibility
17. add(b);add(tf);
18. setSize(300,300);
19. setLayout(**null**);
20. setVisible(**true**);
21. }
22. **public** **void** actionPerformed(ActionEvent e){
23. tf.setText("Welcome");
24. }
25. **public** **static** **void** main(String args[]){
26. **new** AEvent();
27. }
28. }

**public void setBounds(int xaxis, int yaxis, int width, int height);** have been used in the above example that sets the position of the component it may be button, textfield etc.



### 2) Java event handling by outer class

1. **import** java.awt.\*;
2. **import** java.awt.event.\*;
3. **class** AEvent2 **extends** Frame{
4. TextField tf;
5. AEvent2(){
6. //create components
7. tf=**new** TextField();
8. tf.setBounds(60,50,170,20);
9. Button b=**new** Button("click me");
10. b.setBounds(100,120,80,30);
11. //register listener
12. Outer o=**new** Outer(**this**);
13. b.addActionListener(o);//passing outer class instance
14. //add components and set size, layout and visibility
15. add(b);add(tf);
16. setSize(300,300);
17. setLayout(**null**);
18. setVisible(**true**);
19. }
20. **public** **static** **void** main(String args[]){
21. **new** AEvent2();
22. }
23. }
24. **import** java.awt.event.\*;
25. **class** Outer **implements** ActionListener{
26. AEvent2 obj;
27. Outer(AEvent2 obj){
28. **this**.obj=obj;
29. }
30. **public** **void** actionPerformed(ActionEvent e){
31. obj.tf.setText("welcome");
32. }
33. }

### 3) Java event handling by anonymous class

1. **import** java.awt.\*;
2. **import** java.awt.event.\*;
3. **class** AEvent3 **extends** Frame{
4. TextField tf;
5. AEvent3(){
6. tf=**new** TextField();
7. tf.setBounds(60,50,170,20);
8. Button b=**new** Button("click me");
9. b.setBounds(50,120,80,30);
11. b.addActionListener(**new** ActionListener(){
12. **public** **void** actionPerformed(){
13. tf.setText("hello");
14. }
15. });
16. add(b);add(tf);
17. setSize(300,300);
18. setLayout(**null**);
19. setVisible(**true**);
20. }
21. **public** **static** **void** main(String args[]){
22. **new** AEvent3();
23. }
24. }

Next Topic[Java AWT Button](https://www.javatpoint.com/java-awt-button)