Java GUI Programming
AWT/SWING - Event

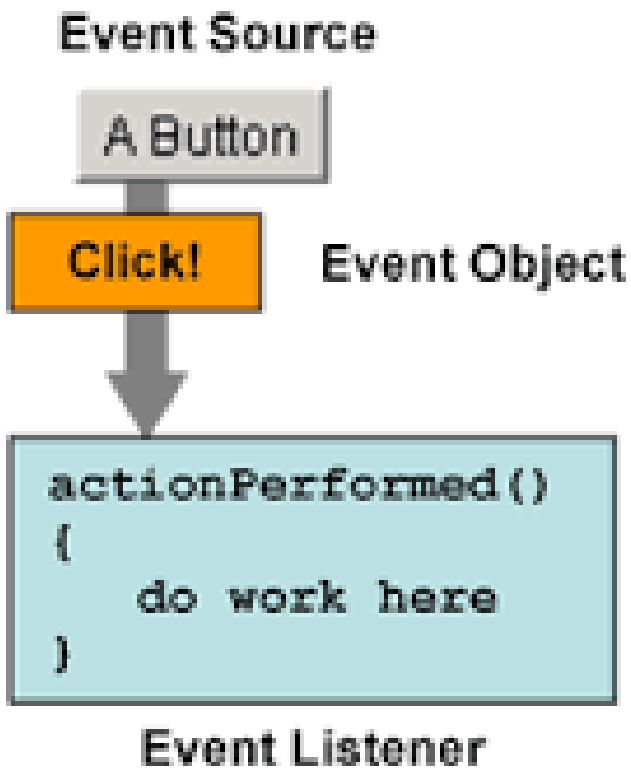
AWT EVENT-DRIVEN PROGRAMING

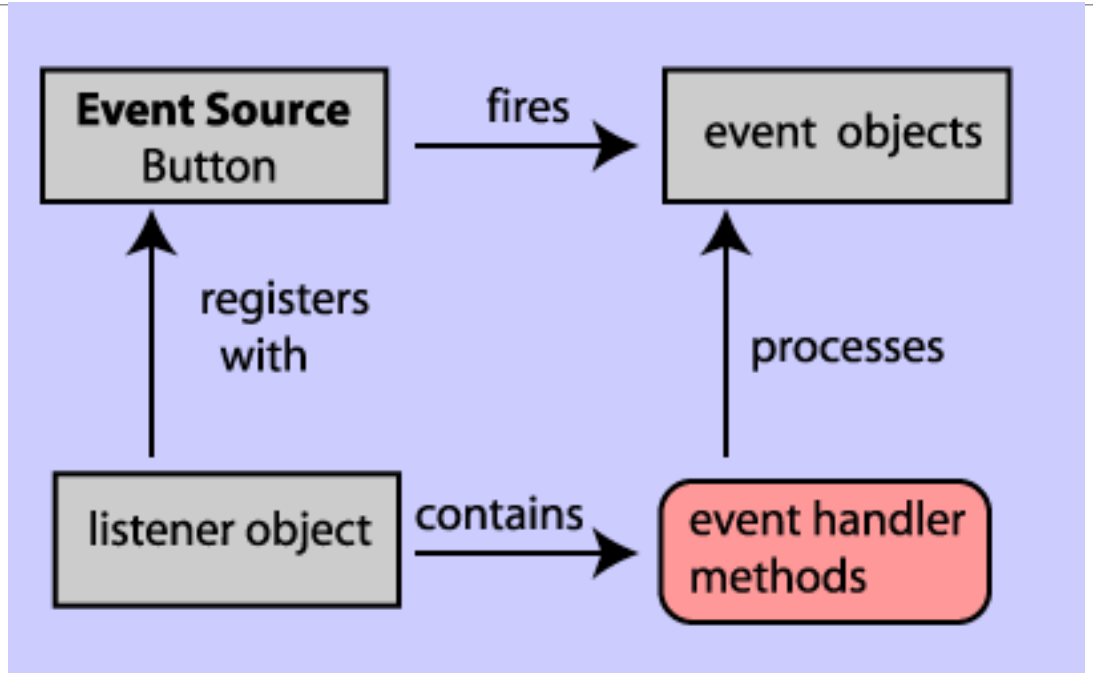
ERIC Y. CHOU, PH.D. IEEE SENIOR MEMBER

Procedural vs. Event-Driven Programming

- Procedural programming is executed in procedural order.
- In event-driven programming, code is executed upon activation of events.

Event-Driven Paradigm in AWT/Swing





- Event Listener also known as Event Handler.
- Object to follow the action is also called Event Target.

Register an Event Listener and Handler

1. Define the Listener Class
2. Create an Listener object
3. Attach the Listener object to button

Source object (e.g., button)

Listener object contains a method for processing the event.

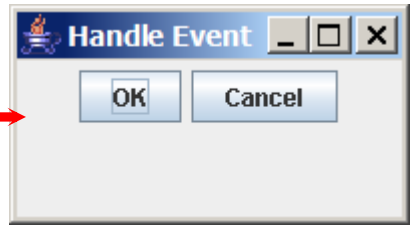
Trace Execution

```
public class HandleEvent extends JFrame {
  public HandleEvent() {
    …
   OKListenerClass listener1 = new OKListenerClass();
    jbtOK.addActionListener(listener1);
    …
  }

  public static void main(String[] args) {
    …
  }
}

class OKListenerClass implements ActionListener {
  public void actionPerformed(ActionEvent e) {
    System.out.println("OK button clicked");
  }
}
```

1. Start from the main method to create a window and display it



Trace Execution

```
public class HandleEvent extends JFrame {
  public HandleEvent() {
    …
   OKListenerClass listener1 = new OKListenerClass();
    jbtOK.addActionListener(listener1);
    …
  }

  public static void main(String[] args) {
    …
  }
}

class OKListenerClass implements ActionListener {
  public void actionPerformed(ActionEvent e) {
    System.out.println("OK button clicked");
  }
}
```

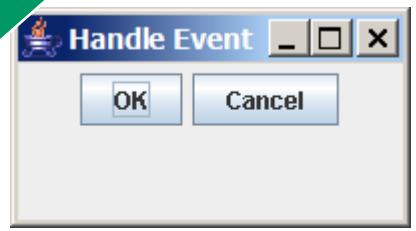
Trace Execution

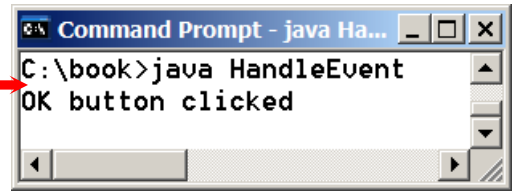
```
public class HandleEvent extends JFrame {
  public HandleEvent() {
    …
   OKListenerClass listener1 = new OKListenerClass();
    jbtOK.addActionListener(listener1);
    …
  }

  public static void main(String[] args) {
    …
  }
}

class OKListenerClass implements ActionListener {
  public void actionPerformed(ActionEvent e) {
    System.out.println("OK button clicked");
  }
}
```

3. Click OK. The JVM invokes the listener's actionPerformed method





Demo Program: HandleEvent.java

Go BlueJ!!!

ActionListener Class and Event Handler

Create an listener object (Listener Object):

```
OKListenerClass listener1 = new OKListenerClass();
```

Add the Listener to button (Event Source):

```
jbtOK.addActionListener(listener1);
```

Listener Class Definition and the Event Handler Method:

```
class OKListenerClass implements ActionListener {
  @Override
  public void actionPerformed(ActionEvent e) {
    System.out.println("OK button clicked");
  }
}
```

Anonymous Inner Class:

```
jbtOK.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent e) {
          System.out.println("OK button clicked");
        }
    }
  );
```

Lambda Expression:

```
 jbtOK.addActionListener( e-> {
          System.out.println("OK button clicked");
         }
     );
```