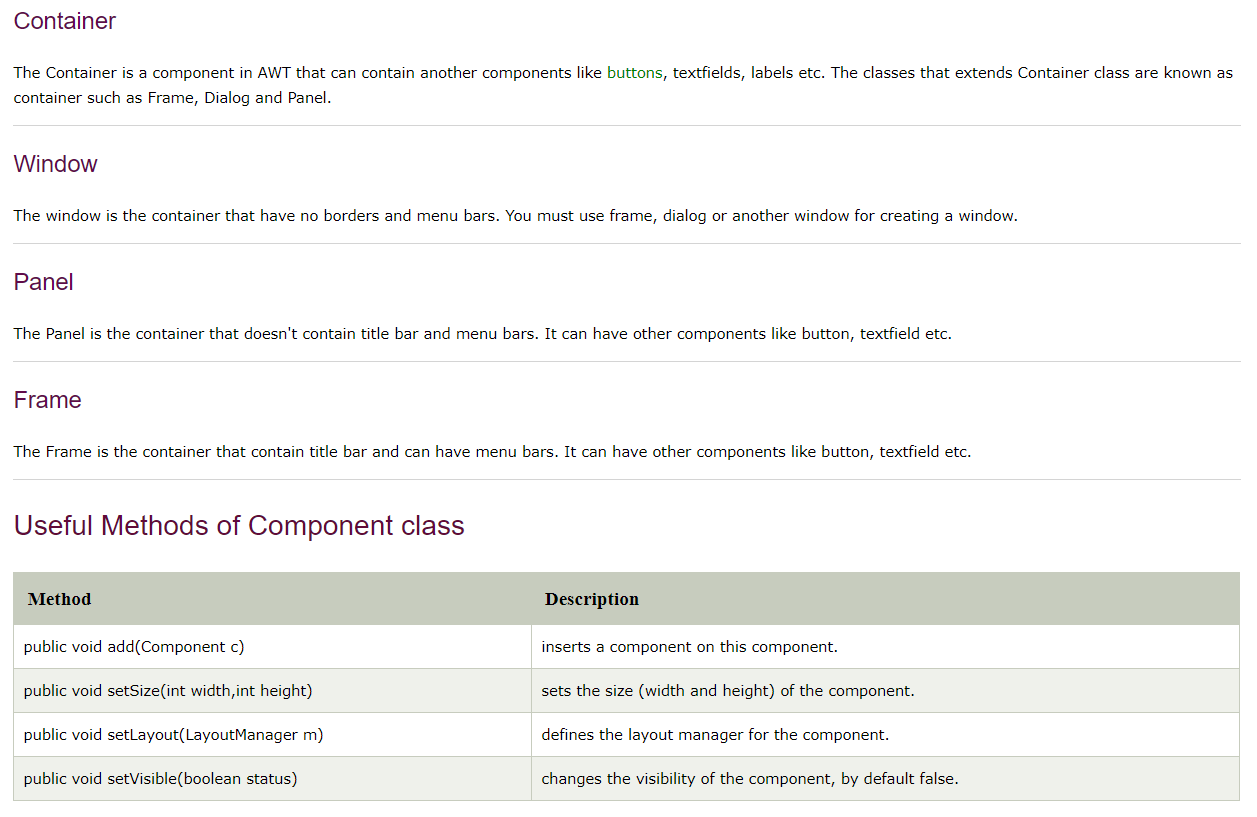
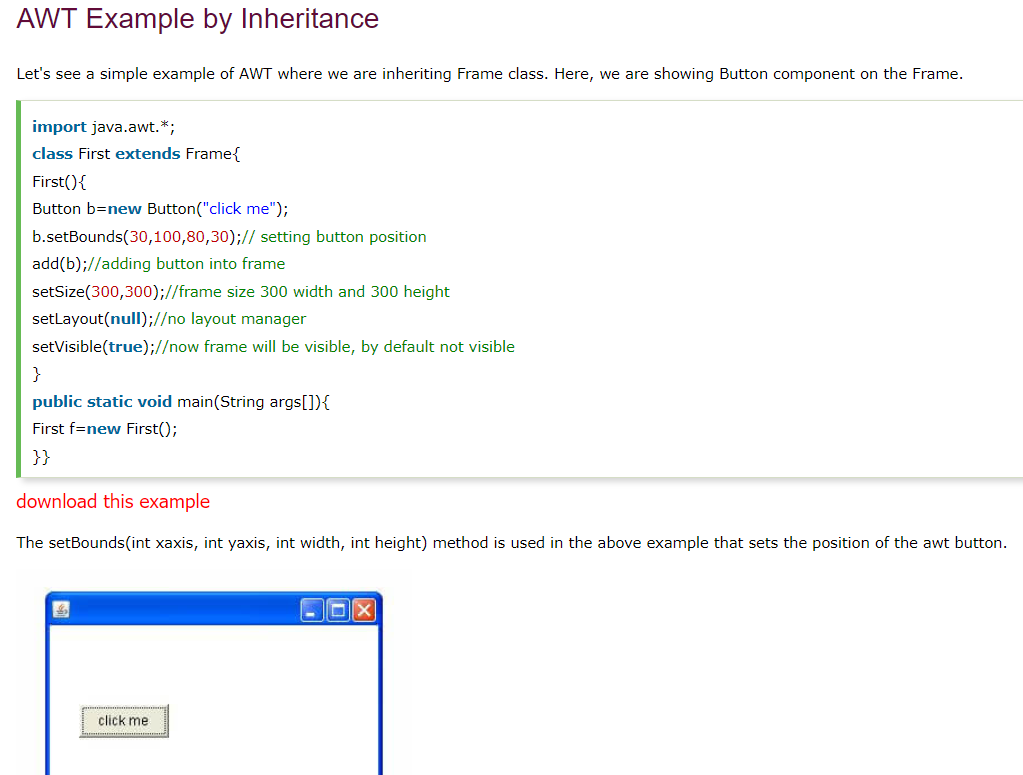
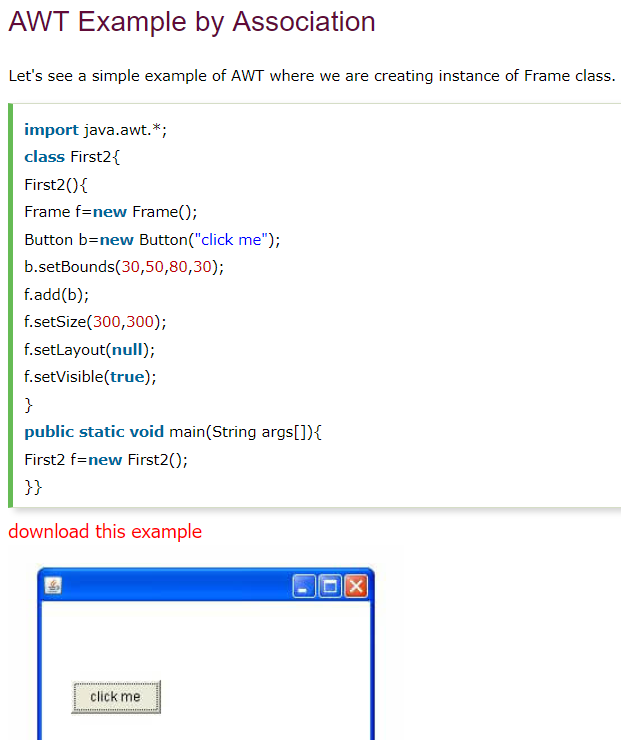
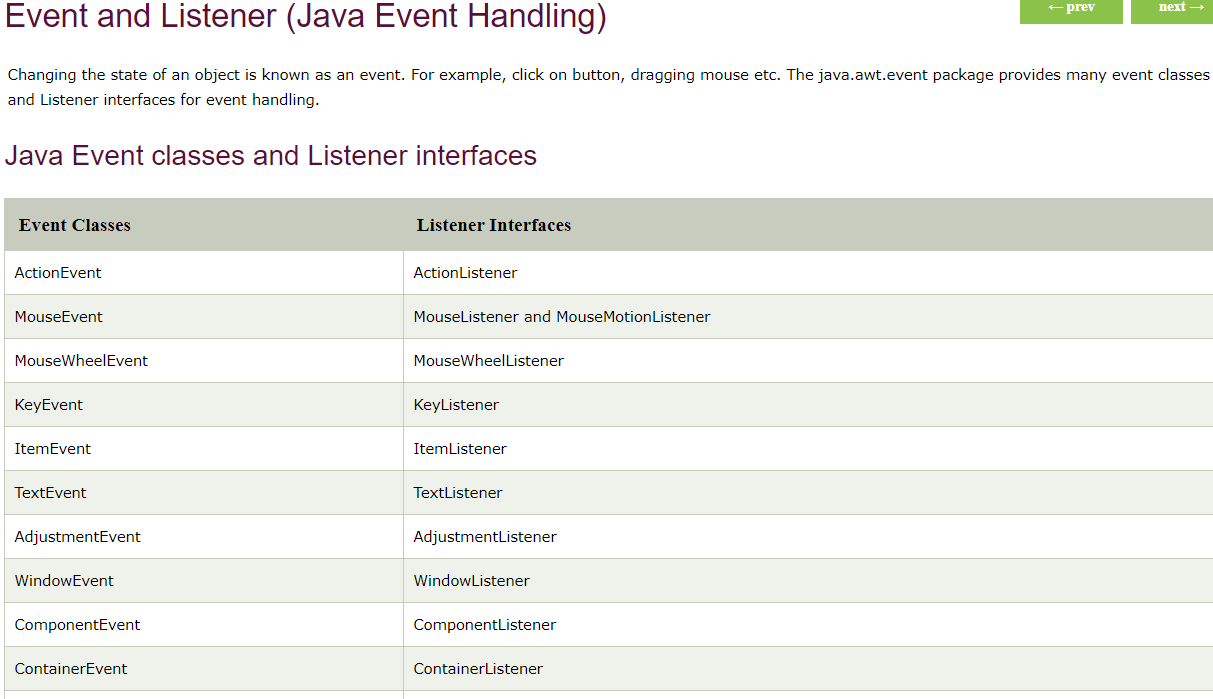
AWT

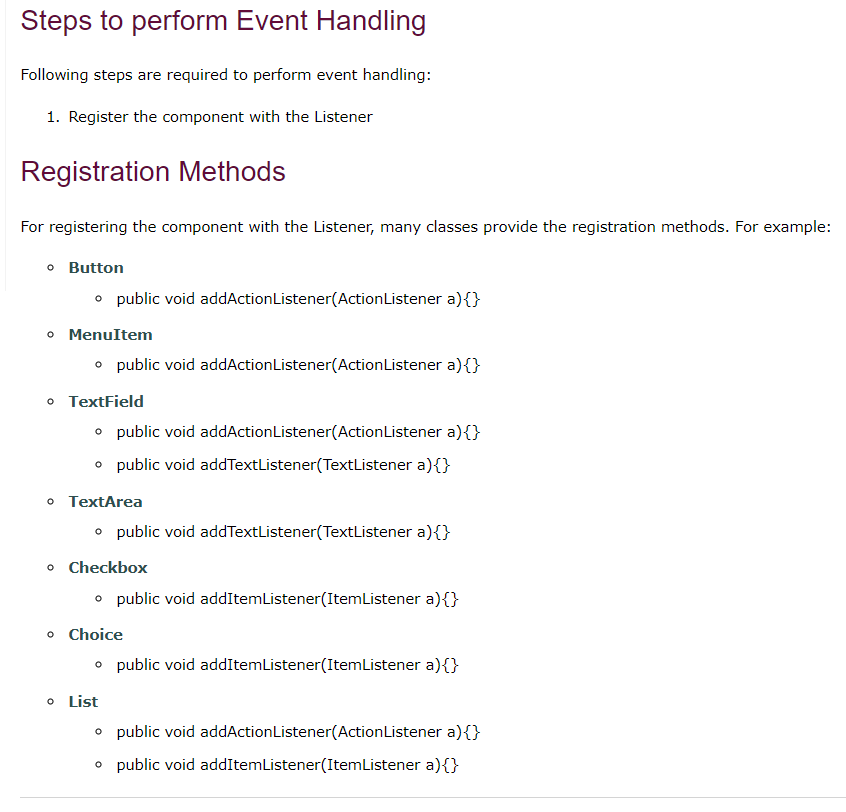
# Heading 1



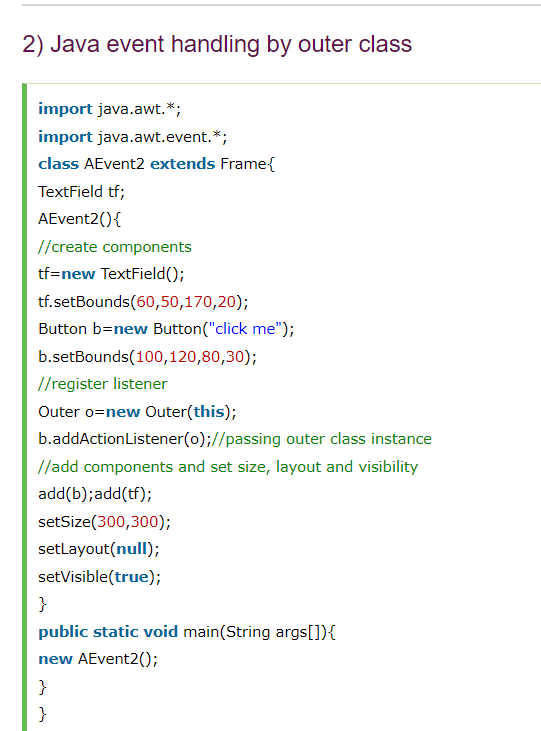


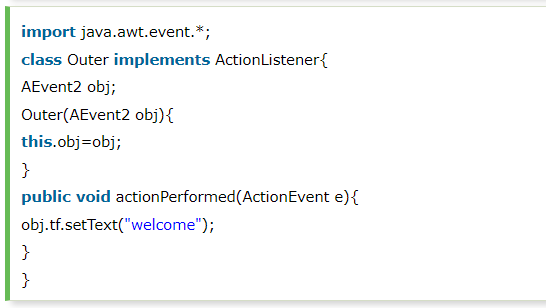












# **Java AWT Button**

# 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.

**public** **class** Button **extends** Component **implements** Accessible

1. **import** java.awt.\*;
2. **public** **class** ButtonExample {
3. **public** **static** **void** main(String[] args) {
4. Frame f=**new** Frame("Button Example");
5. Button b=**new** Button("Click Here");
6. b.setBounds(50,100,80,30);
7. f.add(b);
8. f.setSize(400,400);
9. f.setLayout(**null**);
10. f.setVisible(**true**);
11. }
12. }

## **Java AWT Button Example with ActionListener**

1. **import** java.awt.\*;
2. **import** java.awt.event.\*;
3. **public** **class** ButtonExample {
4. **public** **static** **void** main(String[] args) {
5. Frame f=**new** Frame("Button Example");
6. **final** TextField tf=**new** TextField();
7. tf.setBounds(50,50, 150,20);
8. Button b=**new** Button("Click Here");
9. b.setBounds(50,100,60,30);
10. b.addActionListener(**new** ActionListener(){
11. **public** **void** actionPerformed(ActionEvent e){
12. tf.setText("Welcome to Javatpoint.");
13. }
14. });
15. f.add(b);f.add(tf);
16. f.setSize(400,400);
17. f.setLayout(**null**);
18. f.setVisible(**true**);
19. }
20. }

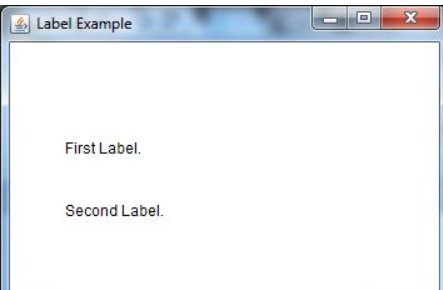
# **Java AWT Label**

The [object](https://www.javatpoint.com/object-and-class-in-java) 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.

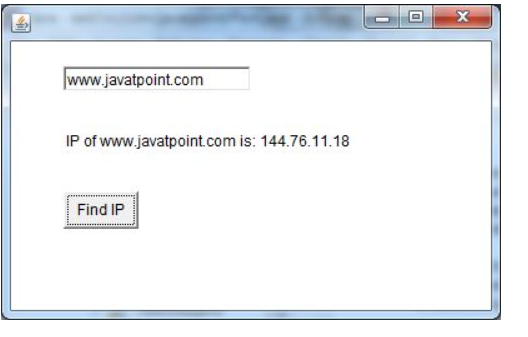
## **AWT Label Class Declaration**

1. **public** **class** Label **extends** Component **implements** Accessible

## **Java Label Example**

1. **import** java.awt.\*;
2. **class** LabelExample{
3. **public** **static** **void** main(String args[]){
4. Frame f= **new** Frame("Label Example");
5. Label l1,l2;
6. l1=**new** Label("First Label.");
7. l1.setBounds(50,100, 100,30);
8. l2=**new** Label("Second Label.");
9. l2.setBounds(50,150, 100,30);
10. f.add(l1); f.add(l2);
11. f.setSize(400,400);
12. f.setLayout(**null**);
13. f.setVisible(**true**);
14. }
15. }  

## **Java AWT Label Example with ActionListener**

1. **import** java.awt.\*;
2. **import** java.awt.event.\*;
3. **public** **class** LabelExample **extends** Frame **implements** ActionListener{
4. TextField tf; Label l; Button b;
5. LabelExample(){
6. tf=**new** TextField();
7. tf.setBounds(50,50, 150,20);
8. l=**new** Label();
9. l.setBounds(50,100, 250,20);
10. b=**new** Button("Find IP");
11. b.setBounds(50,150,60,30);
12. b.addActionListener(**this**);
13. add(b);add(tf);add(l);
14. setSize(400,400);
15. setLayout(**null**);
16. setVisible(**true**);
17. }
18. **public** **void** actionPerformed(ActionEvent e) {
19. **try**{
20. String host=tf.getText();
21. String ip=java.net.InetAddress.getByName(host).getHostAddress();
22. l.setText("IP of "+host+" is: "+ip);
23. }**catch**(Exception ex){System.out.println(ex);}
24. }
25. **public** **static** **void** main(String[] args) {
26. **new** LabelExample();
27.     }
28. }

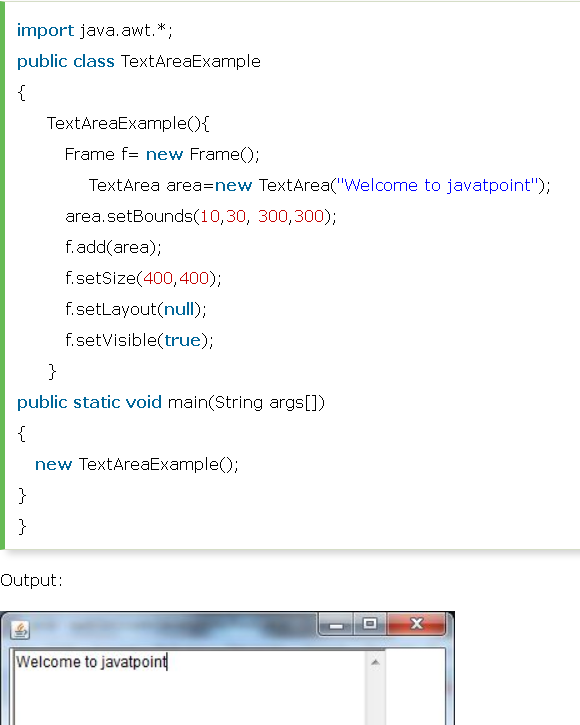
# **Java AWT TextArea**

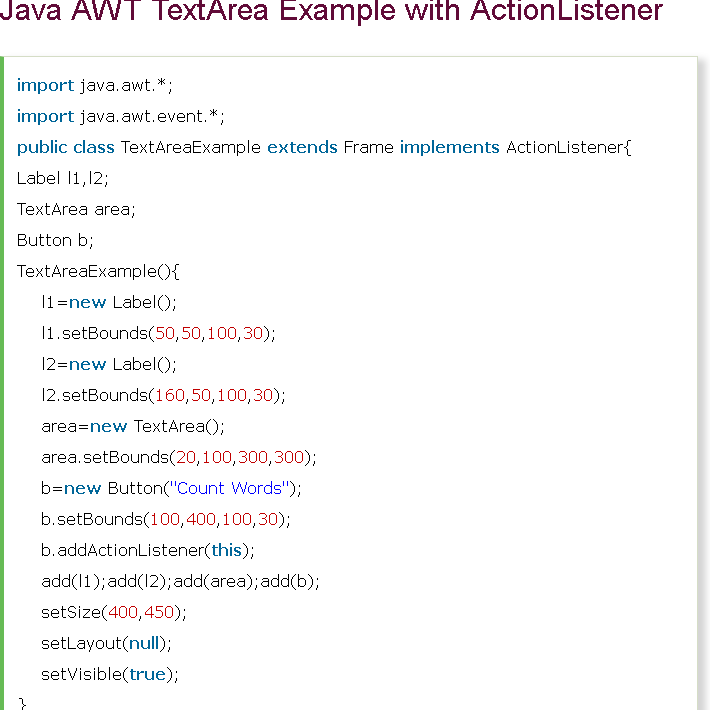
The [object](https://www.javatpoint.com/object-and-class-in-java) of a TextArea class is a multi line region that displays text. It allows the editing of multiple line text. It inherits TextComponent class.

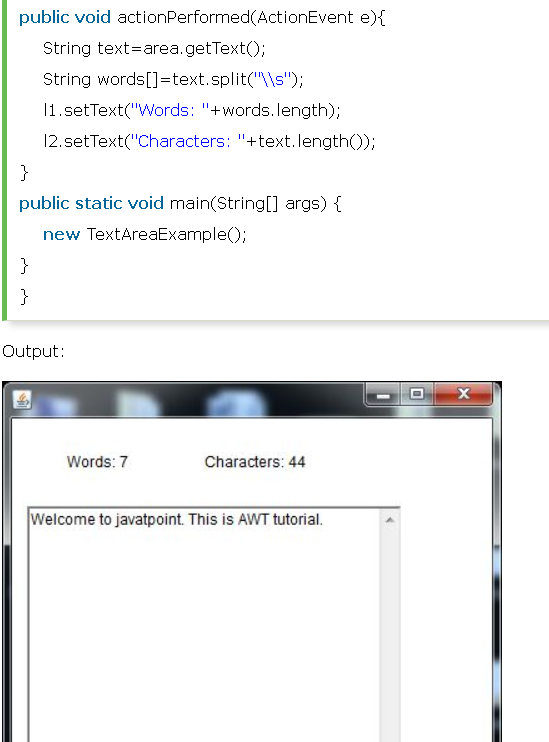
## **AWT TextArea Class Declaration**

1. **public** **class** TextArea **extends** TextComponent

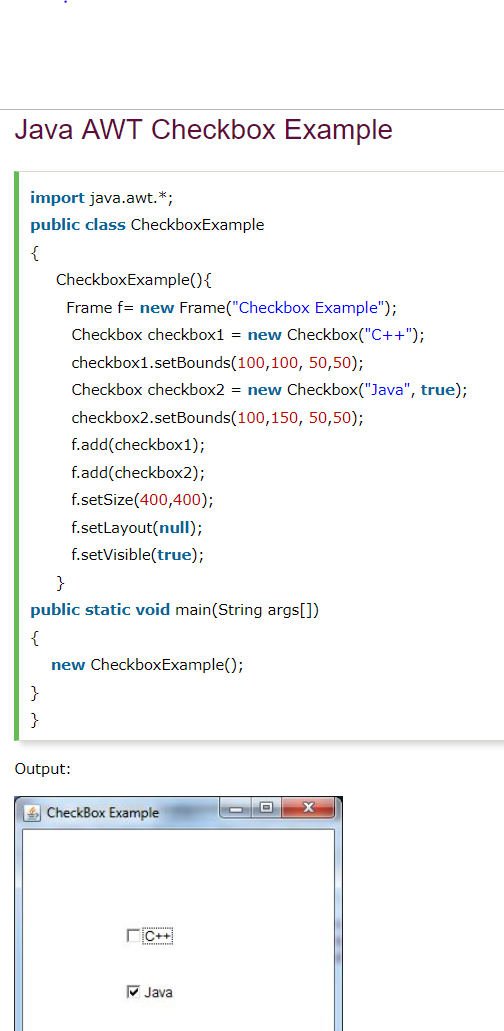
## **Java AWT TextArea Example**







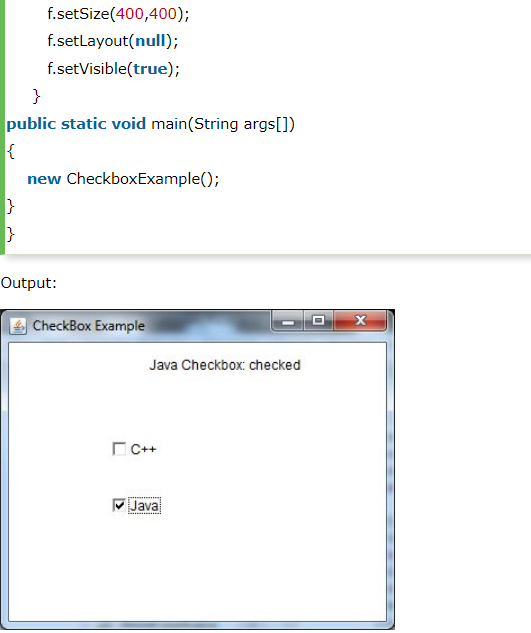
# **Java AWT Checkbox**

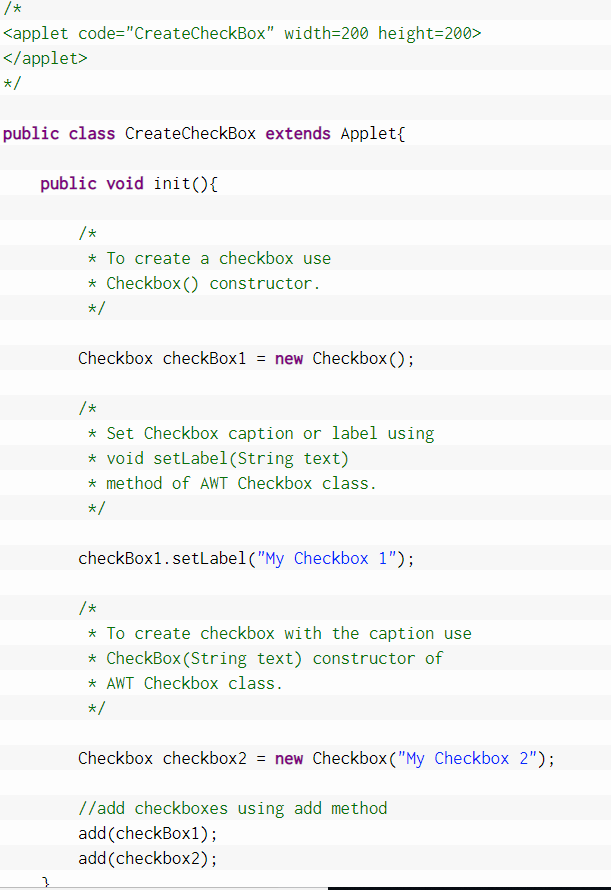
The Checkbox class is used to create a checkbox. It is used to turn an option on (true) or off (false). Clicking on a Checkbox changes its state from "on" to "off" or from "off" to "on".

## **AWT Checkbox Class Declaration**

1. **public** **class** Checkbox **extends** Component **implements** ItemSelectable, Accessible







# **Java AWT CheckboxGroup**

The object of CheckboxGroup class is used to group together a set of [Checkbox](https://www.javatpoint.com/java-awt-checkbox). At a time only one check box button is allowed to be in "on" state and remaining check box button in "off" state. It inherits the [object class](https://www.javatpoint.com/object-class).

#### **Note: CheckboxGroup enables you to create radio buttons in AWT. There is no special control for creating radio buttons in AWT.**

## **AWT CheckboxGroup Class Declaration**

1. **public** **class** CheckboxGroup **extends** Object **implements** Serializable
2. 