Chapter 5

Processing Input with Applets

OBJECTIVES

After you have read and studied this chapter, you should be able to

- Define an applet with multiple methods.
- Incorporate a simple event-handling routine into an applet to process input.
- Construct input-processing applets using Label, TextField, and Button objects from the java.awt package.
- Convert string data to numerical data.
- Use the reserved word this in your programs.
- Run applets without using an applet viewer or browser.

FIGURE 5.1 The applet viewer window when the applet **GreetingApplet** is started.

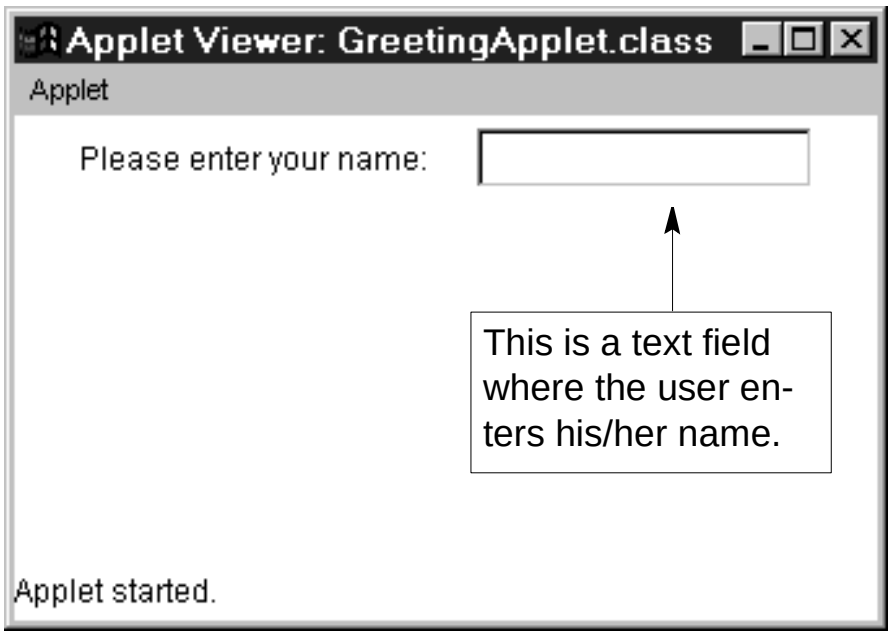


FIGURE 5.2 The applet viewer window after the user enters her name and presses the ENTER key.

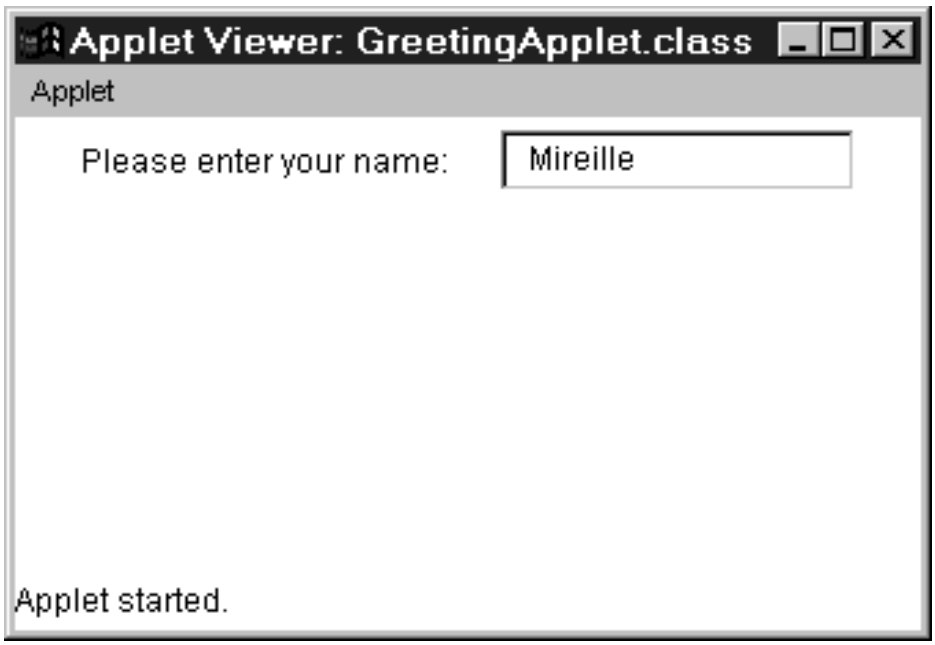


FIGURE 5.3 A template for an applet that is more general than the one we introduced in Chapter 2.

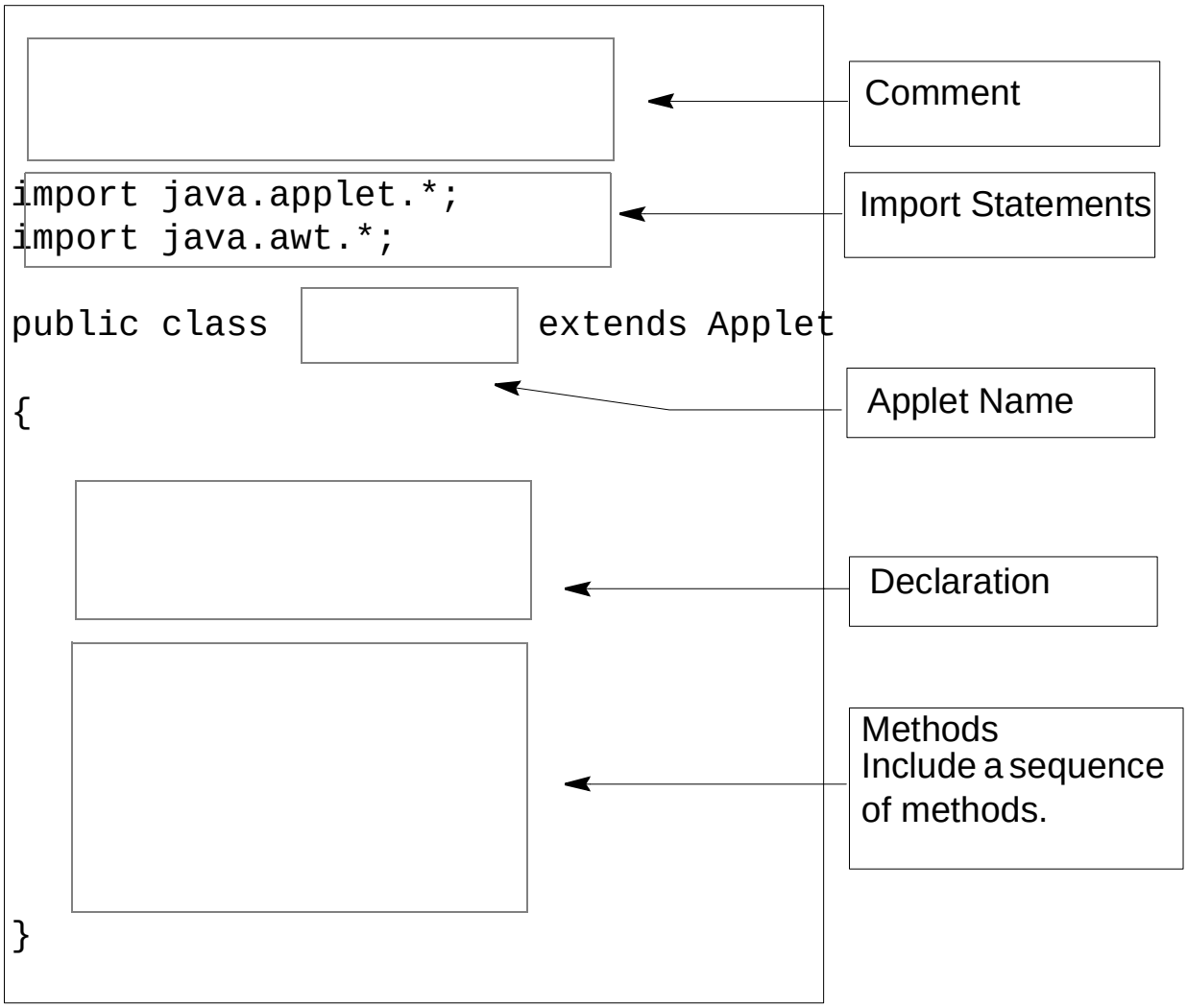


FIGURE 5.4 The applet viewer window with the order of the **add** statements switched.

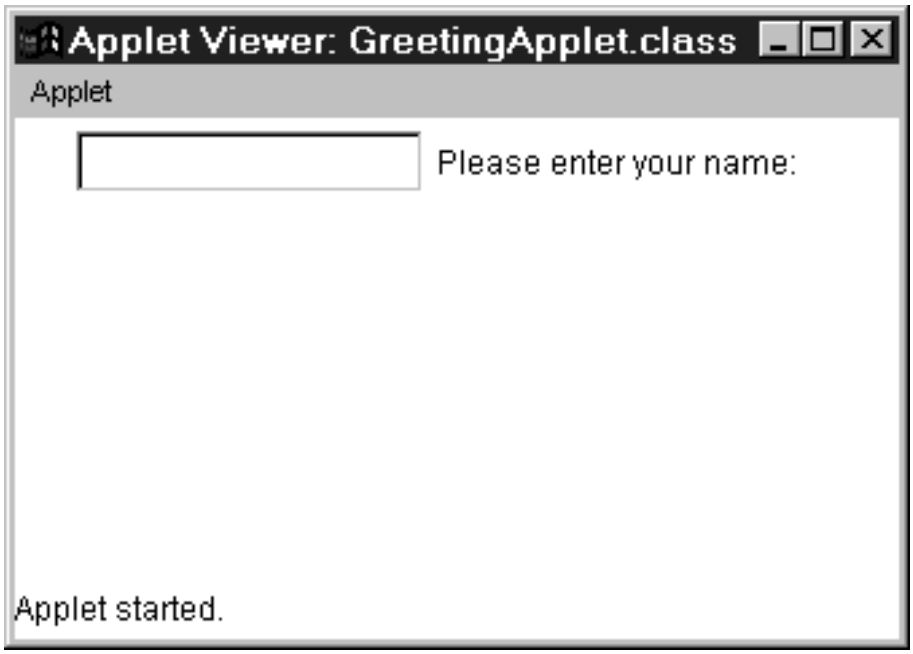
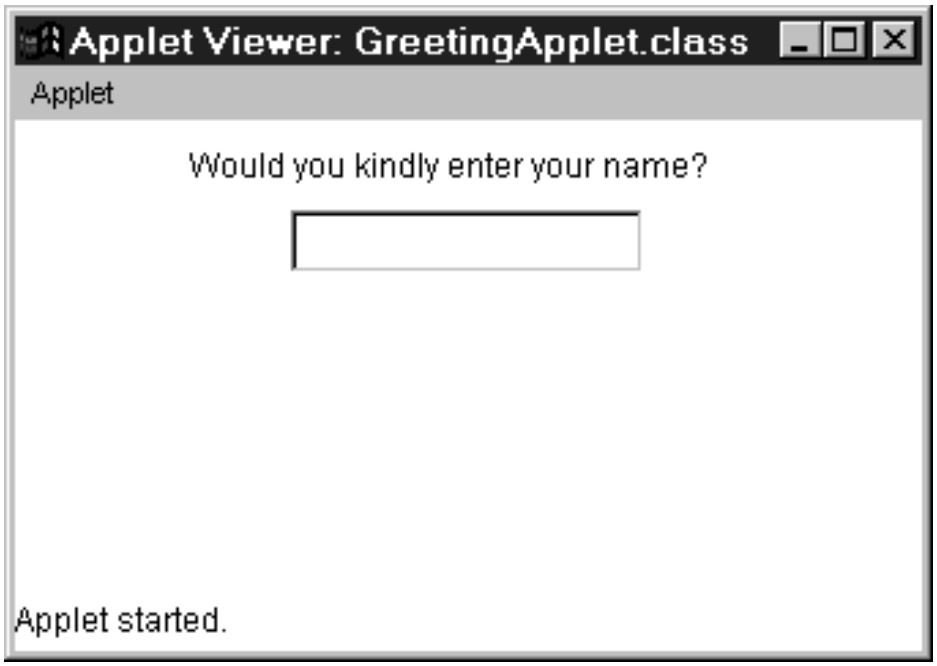


FIGURE 5.5 The applet viewer window with a longer message for the label. Notice that objects in a single row are center-aligned. Alignment becomes more evident if there's only one object in a single row.



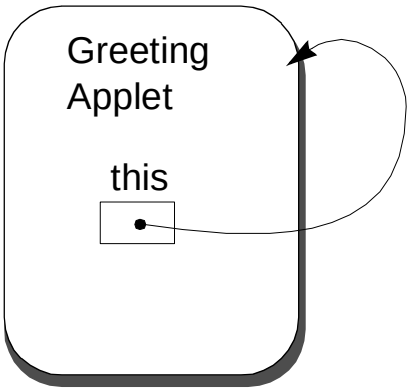


FIGURE 5.6 A template for an input-processing applet. This template is an extension of the one shown in Figure 5.3.

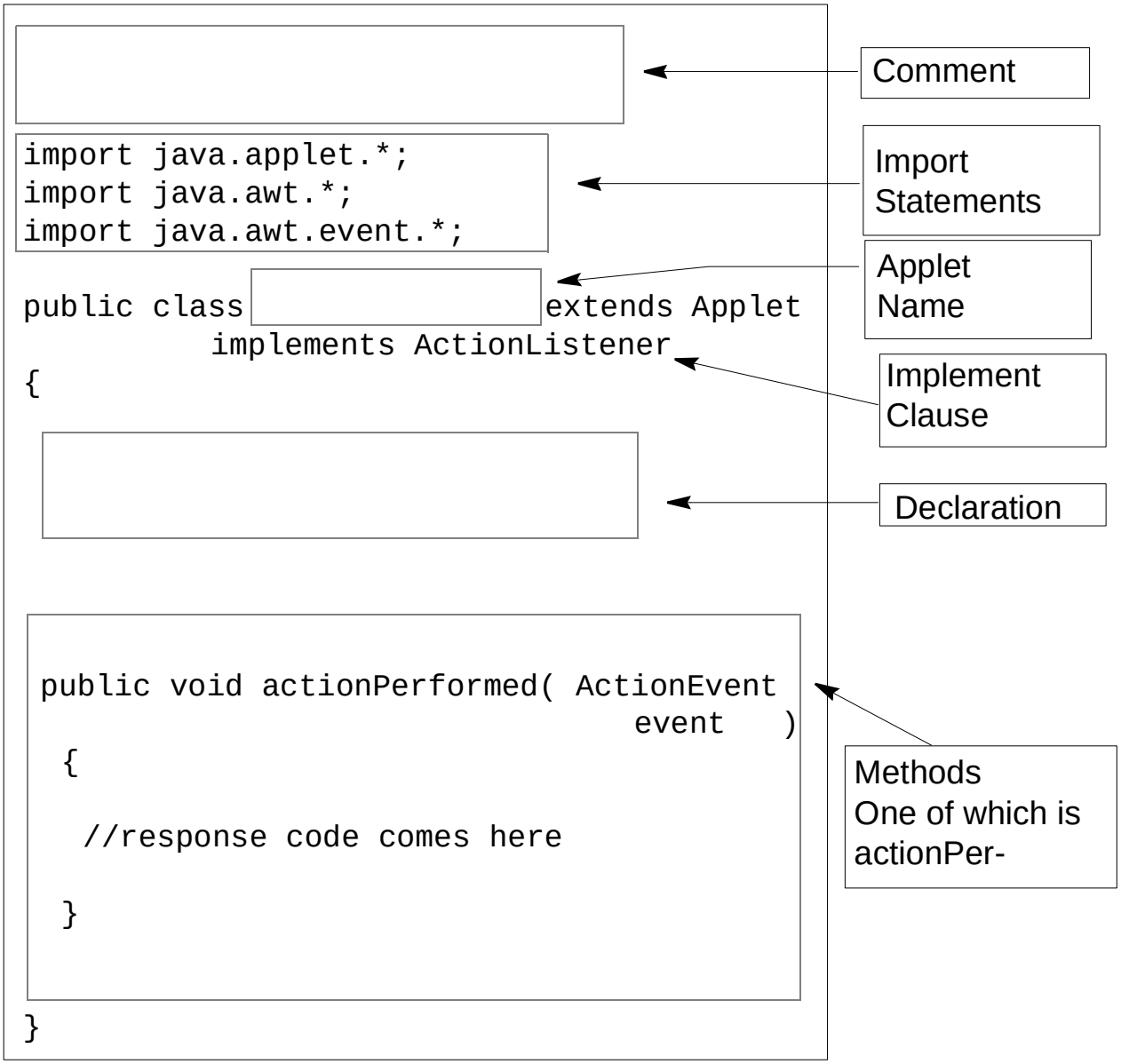
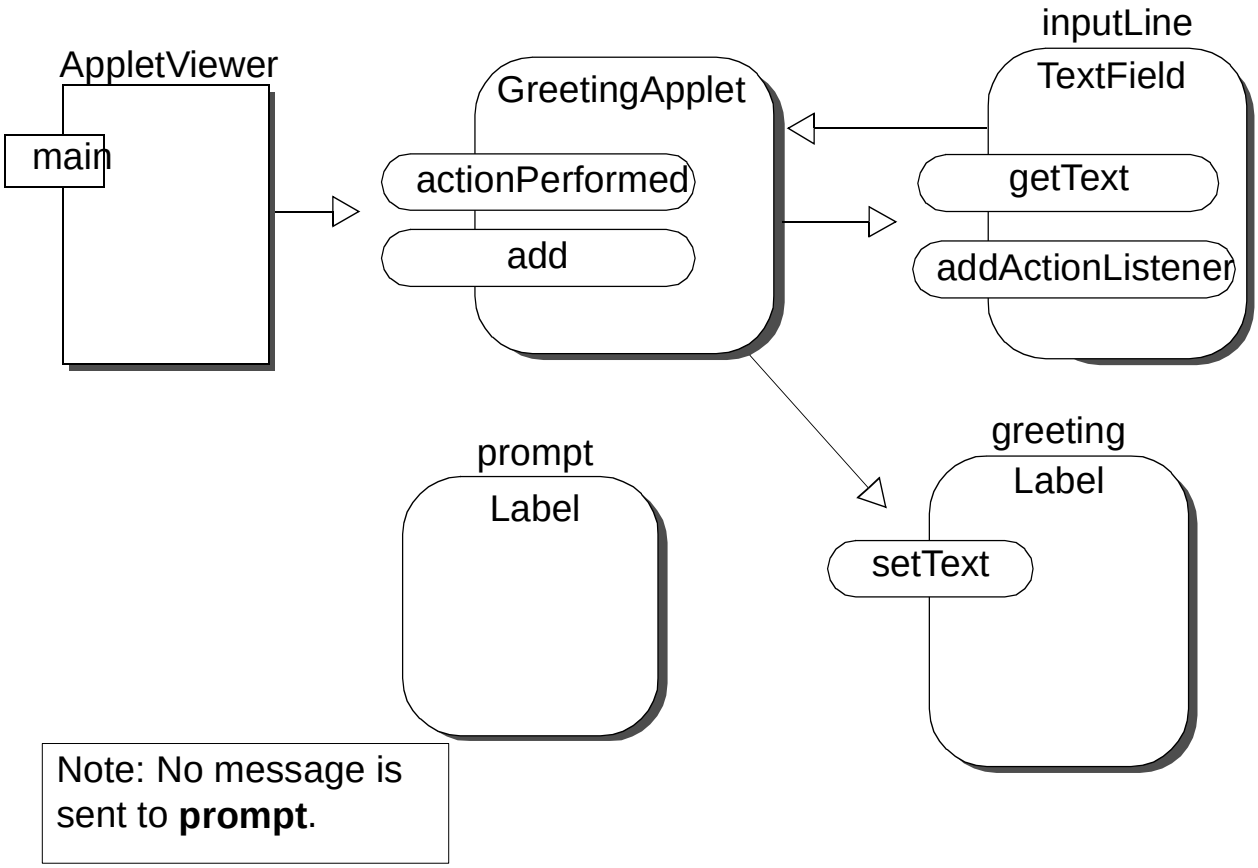
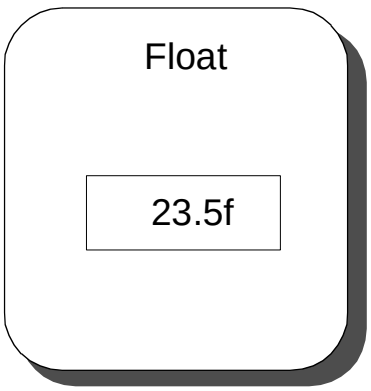


FIGURE 5.7 The object diagram for the **GreetingApplet** program.





A **Float** object containing a **float** value

FIGURE 5.8 The applet with no layout manager and with absolute positioning of GUI objects.

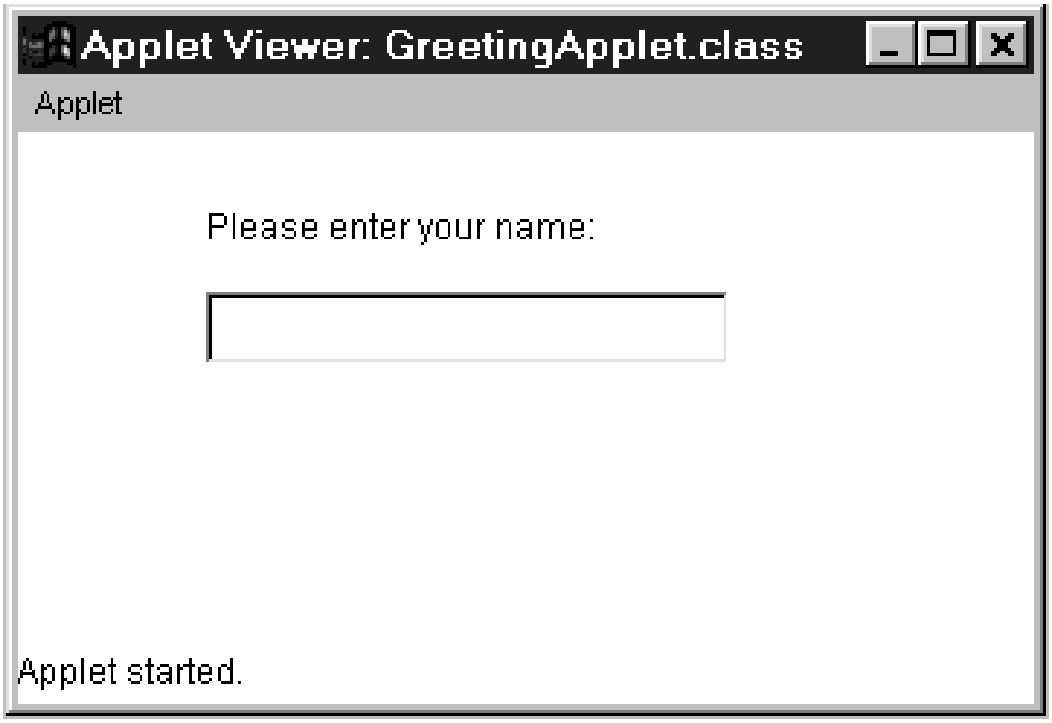


FIGURE 5.9 The **setBounds** method takes four arguments: x, y, width, and height in pixel values .

```
inputLine.setBounds( 65, 55, 150, 25);
```

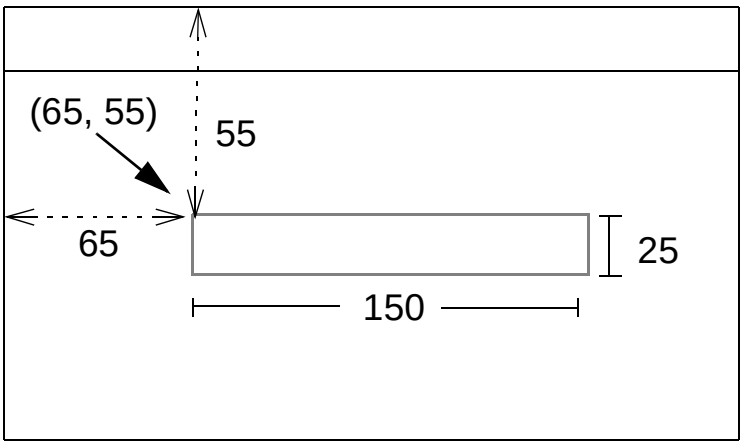


FIGURE 5.10 An applet viewer window running **GreetingAppletWithButton** after the user clicked on the button. This applet uses absolute positioning for laying out the GUI objects.

FIGURE 5.11 The object diagram for the program **BMIApplet**.