11 The Jav aAWT Part I: Mouse Events

11.1

```
public void mouseEntered (MouseEvent e) {
  setMinute(getMinute()+INCREMENT);
  d.clear();
  myClock.display(RADIUS);
}
```

```
  public void mouseExited (MouseEvent e) {
    setMinute(getMinute()+INCREMENT);
    d.clear();
    myClock.display(RADIUS);
  }
```

11.2

```
  public void mouseClicked (MouseEvent e) {
    int x = e.getX(),
        y = e.getY();
    boolean upper = y < HEIGHT/2,
            left = x < WIDTH/2;
    if (upper)
      if (left)       // upper left
        setMinute(getMinute() + 1);
      else            // upper right
        setMinute(getMinute() + 5);
    else if (left)    // lower left
        setMinute(getMinute() + 10);
    else              // lower right
        setMinute(getMinute() + 60);
    d.clear();
    myClock.display(RADIUS);
   }
```

11.3

```
import CSLib.*;
import java.awt.*;
import java.awt.event.*;

public class ClickableClock implements MouseMotionListener {
  // Maintain a clock.
  // Author: Courtney Mickunas, November 21, 2000

  private DrawingBox d;
  private Clock myClock;
  private final int WIDTH = 300,
                    HEIGHT = 320,
                    RADIUS = Math.min(WIDTH, HEIGHT)/2,
                    INCREMENT = 1;

  public ClickableClock () {
    this(12, 0);
  }
  public ClickableClock (int hour, int minute) {
    d = new DrawingBox("ClickableClock Window");
    d.setDrawableSize(WIDTH, HEIGHT);
    myClock = new Clock(d, hour, minute);
    myClock.display(RADIUS);
    d.addMouseMotionListener(this);
  }

  public void mouseMoved (MouseEvent e) {
    setMinute(getMinute()+INCREMENT);
    d.clear();
    myClock.display(RADIUS);
  }
  public void mouseDragged (MouseEvent e) { }

  public void getData () { myClock.getData(); }
  public String toString() { return myClock.toString(); }
  public void setHour (int h) { myClock.setHour(h); }
  public void setMinute (int m) { myClock.setMinute(m); }
  public void set (int hour, int minute) { myClock.set(hour, minute); }
  public int getHour () { return myClock.getHour(); }
  public int getMinute () { return myClock.getMinute(); }
  public boolean priorTo (Clock c) { return myClock.priorTo(c); }
  public boolean after (Clock c) { return c.priorTo(myClock); }
}
```

11.4

If we fail to initialize the cursor in line 12 of the TwoEyes program, then we get a run-time error:

```
Running TwoEyesClient
java.lang.NullPointerException
        at Eyes.compute(Eyes.java:45)
        at Eyes.gaze(Eyes.java:32)
        at TwoEyes.<init>(TwoEyes.java:25)
        at TwoEyesClient.main(TwoEyesClient.java:3)
Exception in thread "main"
```

11.5

```
import CSLib.*;
import java.awt.event.*;

public class Draw implements MouseListener {
  // Draw lines where mouse is clicked
  int last_x = -1, last_y;
  DrawingBox d = new DrawingBox ("Drawing Window");

  public Draw () {
    d.setSize(300, 300);
    d.addMouseListener(this);
    drawClearButton();
    drawSkipButton();
  }

  private void drawClearButton () {
    d.drawRect(0, 0, 40, 40);
    d.drawString("Clear", 6, 25);
  }

  private void drawSkipButton () {
    d.drawRect(0, 40, 40, 40);
    d.drawString("Skip", 6, 65);
  }

  public void mouseClicked (MouseEvent e) {
    int x = e.getX(),
        y = e.getY();
    if (x < 40)
      if (y < 80)
        if (y < 40){ // clear button
          d.clear();
          drawClearButton();
          drawSkipButton();
          last_x = -1;
          return;
        } else {     // skip button
          last_x = -1;
          return;
        }
    if (last_x != -1)
       d.drawLine(last_x, last_y, x, y);
    last_x = x;
    last_y = y;
  }
  public void mousePressed (MouseEvent e) {}
  public void mouseReleased (MouseEvent e) {}
  public void mouseEntered (MouseEvent e) {}
  public void mouseExited (MouseEvent e) {}
}
```

11.6

```
import CSLib.*;
import java.awt.event.*;

public class Draw implements MouseListener {
  // Draw lines where mouse is clicked

  int last_x = -1, last_y, first_x, first_y;
  DrawingBox d = new DrawingBox ("Drawing Window");

  public Draw () {
    d.setSize(300, 300);
    d.addMouseListener(this);
    drawClearButton();
    drawSkipButton();
    drawCloseButton();
  }

  private void drawClearButton () {
    d.drawRect(0, 0, 40, 40);
    d.drawString("Clear", 6, 25);
  }

  private void drawSkipButton () {
    d.drawRect(0, 40, 40, 40);
    d.drawString("Skip", 6, 65);
  }

  private void drawCloseButton () {
    d.drawRect(0, 80, 40, 40);
    d.drawString("Close", 6, 105);
  }

  public void mouseClicked (MouseEvent e) {
    int x = e.getX(),
        y = e.getY();
    if (x < 40) {
      if (y < 120) {   // some button
        if (y < 80) {  // either clear or skip
          if (y < 40){ // clear button
            d.clear();
            drawClearButton();
            drawSkipButton();
            drawCloseButton();
            last_x = -1;
            return;
          } else {     // skip button
            last_x = -1;
            return;
          }
        } else {           // close button
```

```
          d.drawLine(last_x, last_y, first_x, first_y);
          last_x = -1;
          return;
        }
      }
    }
    if (last_x != -1) {
      d.drawLine(last_x, last_y, x, y);
    } else {
      first_x = x;
      first_y = y;
    }
    last_x = x;
    last_y = y;
  }
  public void mousePressed (MouseEvent e) {}
  public void mouseReleased (MouseEvent e) {}
  public void mouseEntered (MouseEvent e) {}
  public void mouseExited (MouseEvent e) {}
}
```