**Video Name:** GUI10

**Topics:**

- Mouse listeners
- pack
- Rectangle contains

**Java Class(es):** package gui_mouse, classes ABox, MouseClickerMain, MouseClickerPanel

Idea: ABox can draw itself. But it doesn’t have paintComponent. Instead, MouseClickerPanel is a JPanel, has a paintComponent method. When repaint() is called, a Graphics object is passed to the paintComponent method of MouseClickerPanel. It in turn calls a draw method of ABox... and passes to the draw method the Graphics object.

**draw method in ABox**

```java
public void draw(Graphics g) {
    g.setColor(color);
    g.fillRect(x, y, BOX_WIDTH, BOX_HEIGHT);
    g.setColor(Color.BLACK);
    g.drawString("Id-"+id, x+BOX_MARGIN, y+BOX_MARGIN);
}
```

**paintComponent method in MouseClickerPanel**

```java
public void paintComponent(Graphics g) {
    super.paintComponent(g);
    for (int i=0; i<boxes.size(); ++i) {
        // We send the Graphics object for the panel into the draw function
        boxes.get(i).draw(g);
    }
}
```

Rather than hard code a size for the JFrame, it’s possible to calculate height/width based on what we have in the panel. So set the size of the JPanel, and then do:

```java
frame.pack();
```

To know whether a mouse is clicked within a rectangular shape on the screen, one technique is the use the `java.awt.Rectangle` class. Create a Rectangle with the location and size desired, then use the `contains` method to determine if the mouse location (determined via the MouseEvent that is passed to each MouseListener method) is within the rectangle.

In this example, the JPanel implements MouseListener. Note that we still need to add the mouse listener to the panel:

```java
addMouseListener(this);
```