We'll now go back to pick up the "graphics track" - the last one or two sections of each chapter.

Many computer programs these days have graphical components:

- Common desktop operating systems have graphical environment
- Pictures, shapes, windows instead of just text
- Mouse as form of input as well as keyboard
Representing Pictures and Color

- A picture consists of pixels, and each pixel is stored separately.
- A black and white picture can be stored using one bit per pixel (0 = white and 1 = black).
- A colored picture requires more information; there are several techniques for representing colors.
- For example, every color can be represented as a mixture of the three additive primary colors Red, Green, and Blue.
- In Java, each color is represented by three numbers between 0 and 255 that collectively are called an *RGB value*.

Coordinate Systems

- Each pixel can be identified using a two-dimensional coordinate system.
- When referring to a pixel in a Java program, we use a coordinate system with the origin in the top-left corner.
## Applets

- A Java application is a stand-alone program with a `main` method (like everything we've seen so far)
- A Java applet is a program that is intended to be delivered over the Web and executed using a web browser
- An applet also can be executed using the appletviewer tool of the Java SDK
- An applet doesn't have a `main` method
- Instead, there are several special methods that serve specific purposes

## Applet is a Superclass

- The class that defines an applet extends the `Applet` class, i.e., is a subclass of `Applet` and inherits all its methods
- See Applet API
- Many of these methods are called from the execution environment (the appletviewer or web browser)
- Your class overrides some of these methods to get the applet to behave the way you want
### Applet Methods

- The `paint` method, for instance, is executed automatically and is used to draw the applet’s contents.
- The `paint` method accepts a parameter that is an object of the `Graphics` class.
- A `Graphics` object defines a *graphics context* on which we can draw shapes and text.
- The `Graphics` class has several methods for drawing shapes.

### Running an Applet

- An applet is embedded into an HTML file using a tag that references the bytecode file of the applet class.
- The bytecode version of the program is transported across the web and executed by a Java interpreter that is part of the browser.
- See `Einstein.java`
The HTML applet Tag

```html
<html>
  <head>
    <title>The Einstein Applet</title>
  </head>
  <body>
    <applet code="Einstein.class" width=350 height=175>
    </applet>
  </body>
</html>
```

Drawing Shapes

- A shape can be filled or unfilled, depending on which method is invoked
- The method parameters specify coordinates and sizes
- Remember that the Java coordinate system has the origin in the top left corner
- Shapes with curves, like an oval, are usually drawn by specifying the shape's *bounding rectangle*
- An arc can be thought of as a section of an oval
**Drawing a Line**

```
page.drawLine (10, 20, 150, 45);
or
page.drawLine (150, 45, 10, 20);
```

**Drawing a Rectangle**

```
page.drawRect (50, 20, 100, 40);
```
**Drawing an Oval**

```java
drawOval (175, 20, 50, 80);
```

**The Color Class**

- A color is defined in a Java program using an object created from the `Color` class.
- The `Color` class also contains several static predefined colors, including:

<table>
<thead>
<tr>
<th>Object</th>
<th>RGB Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color.black</td>
<td>0, 0, 0</td>
</tr>
<tr>
<td>Color.blue</td>
<td>0, 0, 255</td>
</tr>
<tr>
<td>Color.cyan</td>
<td>0, 255, 255</td>
</tr>
<tr>
<td>Color.orange</td>
<td>255, 200, 0</td>
</tr>
<tr>
<td>Color.white</td>
<td>255, 255, 255</td>
</tr>
<tr>
<td>Color.yellow</td>
<td>255, 255, 0</td>
</tr>
</tbody>
</table>
The Color Class

- Every drawing surface has a background color
- Every graphics context has a current foreground color
- Both can be set explicitly
- See Snowman.java (section 2.11)

More Drawing Techniques

- See Bullseye.java
- See Boxes.java
- See BarHeights.java
  (section 3.10)
### Applet Methods

- In previous examples we've used the `paint` method of the `Applet` class to draw on an applet.
- The `Applet` class has several methods that are invoked automatically at certain points in an applet's life.
- The `init` method, for instance, is executed only once when the applet is initially loaded.
- The `start` and `stop` methods are called when the applet becomes active or inactive.
- The `Applet` class also contains other methods that generally assist in applet processing.

### Graphical Objects

- Any object we define by writing a class can have graphical elements.
- The object must simply obtain a graphics context (a `Graphics` object) in which to draw.
- An applet can pass its graphics context to another object just as it can any other parameter.
- See `LineUp.java` (section 4.7)
- See `StickFigure.java` (section 4.7)
An applet is an excellent example of inheritance

Recall that when we define an applet, we extend the Applet class or the JApplet class

The Applet and JApplet classes already handle all the details about applet creation and execution, including:

- interaction with a Web browser
- accepting applet parameters through HTML
- enforcing security restrictions

Our applet classes only have to deal with issues that specifically relate to what our particular applet will do

When we define the paint method of an applet, for instance, we are actually overriding a method defined in the Component class, which is ultimately inherited into the Applet or JApplet class.