javax.swing.BorderFactory

javax.swing.BorderFactory is a toolkit for creating borders around GUI components. Any object of class JComponent can be partially or completely surrounded by a border. GUI programmers generally use borders to enhance the visual clarity of a window.

Example
This GUI window illustrates the use of titled borders to enhance its visual clarity.

You can create many different types of borders, some of which are illustrated below.

<table>
<thead>
<tr>
<th>Some Types of Borders Provided by BorderFactory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raised Bevel Border</td>
</tr>
<tr>
<td><img src="image" alt="Raised Bevel Border" /></td>
</tr>
<tr>
<td>Street Address</td>
</tr>
<tr>
<td>Lowered Bevel Border</td>
</tr>
<tr>
<td><img src="image" alt="Lowered Bevel Border" /></td>
</tr>
<tr>
<td>Street Address</td>
</tr>
<tr>
<td>Rounded Line Border</td>
</tr>
<tr>
<td><img src="image" alt="Rounded Line Border" /></td>
</tr>
<tr>
<td>Street Address</td>
</tr>
<tr>
<td>Matte Border</td>
</tr>
<tr>
<td><img src="image" alt="Matte Border" /></td>
</tr>
<tr>
<td>Street Address</td>
</tr>
</tbody>
</table>
To place a border about a `JComponent` object:

1. Use the appropriate method in the `BorderFactory` class to create a border object.
2. If necessary, you can store the reference to the border object into a reference variable of data type `Border`.
3. Pass the border object to the `setBorder` method of the `JComponent` object.

**Example**
Let’s create the lowered bevel border shown on the address label in the GUI to the right.

Here’s the statement to create the label:

```java
JLabel label = new JLabel( "Street Address" );
```

The statement to create the border:

```java
Border bevel = createBevelBorder( BevelBorder.LOWERED );
```

And the statement to place the border about the label:

```java
label.setBorder( bevel );
```
Class **BorderFactory** is a toolkit like class **Math** – it has no constructors and all its methods are static. Therefore, to make the code in the previous example work, you must import the static objects in **BorderFactory**.

Also, the various border classes (e.g. **Border, BevelBorder**) are in the package **javax.swing.border** so you must import these as well.

**Example**

For the code of the previous example to work, you must import the **BorderFactory** methods and border classes as shown below.

```java
import javax.swing.border.*;
import static javax.swing.BorderFactory.*;
```

There are other types of borders available and each type has attributes that you can control such as the color and thickness of the border. For a full explanation of borders, see the API specification for **BorderFactory** and How to Use Borders in The Java™ Tutorials (google java tutorial borderfactory).
**Exercises**

1. Using the code from the examples, write a complete Java program (application or applet) that builds and displays the GUI shown to the right. The text *Street Address* is in a `JLabel` object having a lowered bevel border.

2. Modify your solution to exercise #1 giving the label a raised bevel border.

3. Modify your solution to exercise #1 giving the label a blue rounded line border whose thickness is 5 pixels.

4. Modify your solution to exercise #1 giving the label a red matte border whose top border has thickness of 1 pixel, the sides are 5 pixels and the bottom is 15 pixels.

5. Modify your solution to exercise #1 giving the label a raised etched border.
6. Modify your solution to exercise #1 giving the label a lowered etched border.

7. Modify your solution to exercise #1 giving the label a compound border that combines an inside matte border with an outside bevel border. The matte border is blue with no top or side borders and a bottom border of 5 pixels. The bevel border is lowered.

8. Write a complete Java program that builds and displays the GUI shown to the right. The GUI component is a `JTextField` with a titled border whose base is a blue line border of 1 pixel thickness.

9. Modify your solution to exercise #8 so that the title lies below the bottom border and is centered horizontally.

10. Modify your solution to exercise #9 so that the title is in an alternate font and color.

11. Modify your solution to exercise #2 of topic *Embellishing Swing Components* so that it places a titled blue line border around the text field. Its title must be *Value* and its thickness 5 pixels.