Programming Assignment 6
Border Crossing Strings

10-point assignment
This assignment gives you another opportunity to create your own supplier class while focusing on string/char manipulation.

When your program begins it must display four bordered text field objects across a ThreeButtonFrame as shown in the image to the right. Note that each bordered text field object has the following characteristics: (a) its overall dimensions are 200 by 60; (b) the text field portion is surrounded by a colored border ten pixels in width; (c) the colored border is surrounded by a 2-pixel wide black border; (d) each corner has a 2-pixel wide diagonal line connecting the text field to the black border. Initially, the four bordered text fields have a cyan-colored border with text pictured to the left. Each bordered text field is located 50 pixels from the nearest side of the ThreeButtonFrame. The upper left bordered text field is 50 pixels down from the title bar; the upper right is 100 pixels down, the lower left is 250 pixels down, and the lower right is 300 pixels down.

Text field objects are similar to label objects, excepting that a text field can be edited at any time during program execution. Below is a class diagram for a class (supplied in the initial program folder) that you should use to create your text fields. The new methods are described briefly below:

<table>
<thead>
<tr>
<th>BigTextField</th>
</tr>
</thead>
<tbody>
<tr>
<td>- int x</td>
</tr>
<tr>
<td>- int y</td>
</tr>
<tr>
<td>- int width</td>
</tr>
<tr>
<td>- int height</td>
</tr>
</tbody>
</table>

«constructor»
+ BigTextField( int x, int y, int w, int h )

«update»
+ void setBounds( int, int, int, int )
+ void setLocation( int, int )
+ void setSize( int, int )
+ void setText( String )
+ void select( int, int )
+ void requestFocusInWindow()
+ void repaint()

«access»
+ int getX()
+ int getY()
+ int getWidth()
+ int getHeight()
+ String getText()

setText(String s)
The displayed text is replaced by the text of String s.

select(int start, int end)
The region of displayed text beginning with the character at position start and extending through to the character at the position before end is highlighted. Note that in the image above the upper left text field has resulted from a call to select(0, 1); Note also that selection is only visible for the text field that has focus. (See requestForFocus().)

requestFocusInWindow()
Modern GUIs operate on a focus system where only one window has focus. The focus window is the one at the forefront and usually displayed in some way to make is appear more prominent. Similarly, only a single text field of the GUI may have focus at any time. Calling requestFocusInWindow() assigns the text field object focus, thus removing focus from the previous focus field. In Java only the text field with focus will highlight its selected text. (Removing focus does not alter the selection, but only the highlighting.)

getText()
Returns the entire text from the text field.
For this assignment you must create a class, called BorderedTextField that implements a bordered text field. Obviously, your Driver will need to create four objects from this class. You are expected to choose methods and instance variables for this supplier class in such a way that minimizes the code in the Driver, reducing the Driver code largely to method calls.

As your program begins execution the first letter of each text field should be selected and the upper left text field should have focus, as shown in the image on the previous page. Below is a description of the required behavior for each button click.

**LEFT**
Clicking the LEFT button should do all of the following:
(a) The border color of the lower left text field is set to cyan.
(b) The first character of the lower left text field is replaced by the character that occurs next in the Unicode character sequence. (i.e., the one with a numeric encoding that is one greater)
(c) The selected region of upper left cell is advanced to the next character (this wraps around to the first character, if the previously selected character was the last one).
(d) The upper left text field receives focus.

**MID**
Clicking the MID button should do all of the following:
(a) The border color of the lower two text fields is set to red.
(b) The selected characters of both of the upper two text fields should be removed (i.e., their prior text is replaced by the text formed from removing the one selected character.)
(c) Once the characters are removed, the selected text is repositioned to the start of the text field, but need not be highlighted.

**RIGHT**
Clicking the RIGHT button should do all of the following:
(a) The border color of the lower right text field is set to cyan.
(b) The first character of the lower right text field is replaced by the character that occurs next in the Unicode character sequence. (i.e., the one with a numeric encoding that is one greater)
(c) The selected region of upper right cell is advanced to the next character (this wraps around to the first character, if the previously selected character was the last one).
(d) The upper right text field receives focus.

You must also remember that at any time during your program’s execution, the user may choose to type new text in any of the text fields. Doing so will alter the focus and text in the field, but nothing else.

**for an three additional two points...**
Modify your program so that for part (b) of the MID button not only is the character removed from the upper two text fields, but also the removed character is appended to the front of the text in the lower text fields. (The removed character in the upper left is appended to the text in the lower left and the removed character in the upper right is appended to the text in the lower right.

**to submit your solution...**
There is an original folder on Riley’s website that you should download, and an example program to see the desired behavior. Email your solution in the form of an attachment to riley@cs.uwlax.edu. Please compress the entire folder into a single zip file and attach this zip file to the email.

**Due date:** Nov. 5, 2010