Still can submit (with slip day) PSA2!
Start PSA3 😊

How was it?
A. No sweat!
B. Not too bad
C. So-so
D. Not so good
E. Totally awful
Now on gradesource:

- Integrity Agreement: If you don’t sign it you won’t get MAGIC NUMBER
- Participation and reading quizzes, week 2
- Slip days USED

<table>
<thead>
<tr>
<th>Academic Integrity Agreement</th>
<th>P4_8</th>
<th>P4_10</th>
<th>RQ4_8</th>
<th>RQ4_10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>Score</td>
<td>Score</td>
<td>Score</td>
<td>Score</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>1*</td>
<td>1</td>
<td>4*</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>1*</td>
<td>1</td>
<td>4</td>
<td>3*</td>
</tr>
<tr>
<td>1</td>
<td>1*</td>
<td>1</td>
<td>2*</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>1*</td>
<td>1</td>
<td>4</td>
<td>3*</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>*</td>
<td>1</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>1*</td>
<td>1</td>
<td>4*</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>1*</td>
<td>1</td>
<td>3*</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>1*</td>
<td>1</td>
<td>4*</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>1*</td>
<td>1</td>
<td>3</td>
<td>1*</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>*</td>
<td>3</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>*</td>
<td>3</td>
<td>*</td>
</tr>
</tbody>
</table>
Seating Layout – Check your group!

**CENTRAL 212 - SECTION A00 8B**
Peer Instruction Layout:

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
<tr>
<td>G</td>
<td>H</td>
<td>I</td>
<td>J</td>
<td>K</td>
<td>L</td>
</tr>
</tbody>
</table>

**CENTRAL 105 - SECTION B00 8B**
Peer Instruction Layout:

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
<tr>
<td>G</td>
<td>H</td>
<td>I</td>
<td>J</td>
<td>K</td>
<td>L</td>
</tr>
</tbody>
</table>
public static void main(String[] args) {
    JFrame myFrame = new JFrame("This is my window");
    myFrame.setSize(300, 400);

    FlowLayout flow = new FlowLayout(FlowLayout.LEFT);
    GridLayout grid = new GridLayout(5, 6);
    myFrame.setLayout(flow);
    for (int i = 0; i < 20; i++) {
        JLabel label = new JLabel("CSE 8B");
        label.setFont(new Font("Serif", Font.ITALIC, i*3));
        myFrame.add(label);
    }

    myFrame.setVisible(true);
}

Sketch (more or less) what you think this GUI will look like
Layout Managers

Sketch (more or less) what you think this GUI will look like
Concept - videos

- TUTOR VIDEOS:
  
  *e.g. Looking at arrays through memory models*
  
  [https://www.youtube.com/watch?v=WIdXkFY-4yg](https://www.youtube.com/watch?v=WIdXkFY-4yg)

  What do you want to see there?!

  A. FOR EACH loop
  B. How to use the [http://docs.oracle.com](http://docs.oracle.com) documentation?
  C. Throwing Exceptions
  D. WRITE a File from your program
  E. Others ... (POST ON PIAZZA!)
What are those “Constant” values?

“Typical pre-defined” values are defined as static variables in the Class:

```java
public class Color implements Paint, Serializable{
    // Color.java
    
    
    /** Constant for the color white: R=255, G=255, B=255. */
    public static final Color white = new Color(0xffffff, false);

    public static final Color WHITE = white;

    /** Constant for the color light gray: R=192, G=192, B=192. */
    public static final Color lightGray = new Color(0xc0c0c0, false); 

    . . .

    public class Font implements Serializable{
        // Font.java
        
        /** * Constant indicating a "plain" font. */
        public static final int PLAIN = 0;

        /** * Constant indicating a "bold" font. */
        public static final int BOLD = 1;

        /** * Constant indicating an "italic" font. */
        public static final int ITALIC = 2;

        . . .
```
Sort and Search methods in Arrays

(from docs.oracle.com)

**sort**

```java
public static void sort(byte[] a)
```

Sorts the specified array into ascending numerical order.

*Implementation note:* The sorting algorithm is a Dual-Pivot Quicksort by Vladimir Yaroslavskiy, Jon Bentley, and Joshua Bloch. This algorithm offers $O(n \log(n))$ performance on many data sets that cause other quicksorts to degrade to quadratic performance, and is typically faster than traditional (one-pivot) Quicksort implementations.

**Parameters:**

- `a` - the array to be sorted
bin\ary\Search

public static int binarySearch(int[] a, int key)

*Searches the specified array of ints for the specified value using the binary search algorithm.* The array must be sorted (as by the `sort(int[])` method) prior to making this call. If it is not sorted, the results are undefined. If the array contains multiple elements with the specified value, there is no guarantee which one will be found.

**Parameters:**
a - the array to be searched
tkey - the value to be searched for

**Returns:**
index of the search key, if it is contained in the array;
otherwise, `(-insertion point) - 1`). The **insertion point** is defined as the point at which the key would be inserted into the array: the index of the first element greater than the key, or `a.length` if all elements in the array are less than the specified key. Note that this guarantees that the return value will be `>= 0` if and only if the key is found.
public static String rc(String x, char y, char z) {
    char[] s = s.toCharArray();
    char y2 = Character.toLowerCase(y);  char z3 =
    Character.toLowerCase(z);  char y4 = Character.toUpperCase(y);
    char z5 = Character.toUpperCase(z);
    for (int c = 0; c < s.length; c++) {
        if (chars[i] == y2)
            chars[c] = z3;  else if(chars[c] == y4) chars[c] = z5;
    }
    return String.valueOf(s);
}

What does this method do??
A. I have no idea
B. I think I might know
C. I am (pretty) sure I know
public static String replaceChar( String s, char gone, char here )
{
    char[] chars = s.toCharArray();
    char goneLow = Character.toLowerCase(gone);
    char hereLow = Character.toLowerCase(here);
    char goneUp = Character.toUpperCase(gone);
    char hereUp = Character.toUpperCase(here);

    for ( int i = 0; i < chars.length; i++ ) {
        if ( chars[i] == goneLow )
            chars[i] = hereLow;
        else if(chars[i] == goneUp)
            chars[i] = hereUp;
    }
    return String.valueOf( chars );
}

What does this method do??
A. I have no idea
B. I think I might know
C. I am (pretty) sure I know
A note about style...

/** Description . . . */
public static String replaceChar (String s, char gone, char here)
{
    char[] chars = s.toCharArray();
    char goneLow = Character.toLowerCase(gone);
    char hereLow = Character.toLowerCase(here);
    char goneUp = Character.toUpperCase(gone);
    char hereUp = Character.toUpperCase(here);

    for ( int i = 0; i < chars.length; i++ )
    {
        if ( chars[i] == goneLow ){
            chars[i] = hereLow;
        }
        else if(chars[i] == goneUp){
            chars[i] = hereUp;
        }
    }

    return String.valueOf( chars );
}

Style guidelines on the CSE8B homepage. Starting NOW (with PSA3) style matters!
A note about style…

/** Description . . . */
public static String replaceChar (String s, char gone, char here)
{
    char[] chars = s.toCharArray();
    char goneLow = Character.toLowerCase(gone);
    char hereLow = Character.toLowerCase(here);
    char goneUp = Character.toUpperCase(gone);
    char hereUp = Character.toUpperCase(here);
    for (int i = 0; i < chars.length; i++)
    {
        if (chars[i] == goneLow)
            chars[i] = hereLow;
        else if(chars[i] == goneUp)
            chars[i] = hereUp;
    }
    return String.valueOf(chars);
}

And don’t forget your header comment!!

Style guidelines on the CSE8B homepage. Starting NOW (with PSA3) style matters!
PSA 3:
carefully read **notes and hints** at the PSA website

**PART 1: COUNT WORDS**
getWordsFromFile: read all words from a file:

```java
```

setUniqueAndCounts: This method will need to create two new arrays: one for counts (int[]) and one for uniqueWords (String[]).

Don’t forget:

```java
import ... 
```

```java
public String[] getWordsFromFile( String filename ) throws IOException
```

**PART 2: DISPLAY WORDS** in a SIMPLE GUI