Enumeration

public static final int FROSH = 1;
public static final int SOPHOMORE = 2;
public static final int JUNIOR = 3;
public static final int SENIOR = 4;
public static final int OTHER = 5;

private int askForYear() {
    while (true) {
        int year = readInt("Enter class year: ");
        if (year >= FROSH && year <= OTHER) return year;
    }
}
Converting Enumeration

private String getYearAsString(int year) {
    switch (year) {
        case FROSH:
            return "Frosh";
        case SOPHOMORE:
            return "Sophomore";
        case JUNIOR:
            return "Junior";
        case SENIOR:
            return "Senior";
        default:
            return "Other";
    }
}
The letter A, for example, has the Unicode value 1018, which is the sum of the row and column labels.

Using portions of slides by Eric Roberts
Using Methods of `Character`

```java
public void run() {
    String str = readLine("Line: ");

    char ch = str.charAt(0);
    println("Original first char: " + ch);

    ch = Character.toUpperCase(ch);
    println("Uppercase first char: " + ch);
}
```
## Useful Methods in the `Character` Class

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>static boolean isDigit(char ch)</code></td>
<td>Determines if the specified character is a digit.</td>
</tr>
<tr>
<td><code>static boolean isLetter(char ch)</code></td>
<td>Determines if the specified character is a letter.</td>
</tr>
<tr>
<td><code>static boolean isLetterOrDigit(char ch)</code></td>
<td>Determines if the specified character is a letter or a digit.</td>
</tr>
<tr>
<td><code>static boolean isLowerCase(char ch)</code></td>
<td>Determines if the specified character is a lowercase letter.</td>
</tr>
<tr>
<td><code>static boolean isUpperCase(char ch)</code></td>
<td>Determines if the specified character is an uppercase letter.</td>
</tr>
<tr>
<td><code>static boolean isWhitespace(char ch)</code></td>
<td>Determines if the specified character is whitespace (spaces and tabs).</td>
</tr>
<tr>
<td><code>static char toLowerCase(char ch)</code></td>
<td>Converts <code>ch</code> to its lowercase equivalent, if any. If not, <code>ch</code> is returned unchanged.</td>
</tr>
<tr>
<td><code>static char toUpperCase(char ch)</code></td>
<td>Converts <code>ch</code> to its uppercase equivalent, if any. If not, <code>ch</code> is returned unchanged.</td>
</tr>
</tbody>
</table>

*Using portions of slides by Eric Roberts*
Working with **String**

```java
public void run() {
    String s1 = "CS106";
    String s2 = "A";
    String s3 = "I got an " + s2 + " in " + s1 + s2;

    println(s3);
}
```

I got an A in CS106A
# Useful Methods in the **String** Class

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>int length()</code></td>
<td>Returns the length of the string</td>
</tr>
<tr>
<td><code>char charAt(int index)</code></td>
<td>Returns the character at the specified index. Note: Strings indexed starting at 0.</td>
</tr>
<tr>
<td><code>String substring(int p1, int p2)</code></td>
<td>Returns the substring beginning at <code>p1</code> and extending up to but not including <code>p2</code></td>
</tr>
<tr>
<td><code>String substring(int p1)</code></td>
<td>Returns substring beginning at <code>p1</code> and extending through end of string.</td>
</tr>
<tr>
<td><code>boolean equals(String s2)</code></td>
<td>Returns true if string <code>s2</code> is equal to the receiver string. This is case sensitive.</td>
</tr>
<tr>
<td><code>int compareTo(String s2)</code></td>
<td>Returns integer whose sign indicates how strings compare in lexicographic order</td>
</tr>
<tr>
<td><code>int indexOf(char ch)</code> or <code>int indexOf(String s)</code></td>
<td>Returns index of first occurrence of the character or the string, or -1 if not found</td>
</tr>
<tr>
<td><code>String toLowerCase()</code> or <code>String toUpperCase()</code></td>
<td>Returns a lowercase or uppercase version of the receiver string</td>
</tr>
</tbody>
</table>

*Using portions of slides by Eric Roberts*
This program reverses a string.
Enter a string: STRESSED
STRESSED spelled backwards is DESSERTS