CSE11 Discussion 2

with Chris Taylor
“Constructors”, how you can set up the variable (not super important yet)

```
Scanner scnr = new Scanner(System.in);
snr.close();
scnr.hasNext();
```

“Methods”, the functions available

<table>
<thead>
<tr>
<th>Method Summary</th>
<th>Method and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>close()</td>
<td>Closes this scanner.</td>
</tr>
<tr>
<td>delimeter()</td>
<td>Returns the Pattern this Scanner is searching for.</td>
</tr>
<tr>
<td>findInLine(Pattern pattern, String source)</td>
<td>Attempts to find the next occurrence of the specified Pattern in the specified string.</td>
</tr>
<tr>
<td>findInLine(String pattern, String source)</td>
<td>Attempts to find the next occurrence of the specified string in the specified string.</td>
</tr>
<tr>
<td>findWithinHorizon(Pattern pattern, String source)</td>
<td>Attempts to find the next occurrence of the specified Pattern in the specified string, starting at the specified offset.</td>
</tr>
<tr>
<td>findWithinHorizon(String pattern, String source)</td>
<td>Attempts to find the next occurrence of the specified string in the specified string, starting at the specified offset.</td>
</tr>
<tr>
<td>hasNext()</td>
<td>Returns true if this Scanner has more tokens to return.</td>
</tr>
<tr>
<td>hasNext(Pattern pattern)</td>
<td>Returns true if this Scanner has a token that matches the specified Pattern.</td>
</tr>
<tr>
<td>hasNext(String pattern)</td>
<td>Returns true if this Scanner has a token that matches the specified string.</td>
</tr>
</tbody>
</table>
# Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void</code></td>
<td><strong>moveTo(int x, int y)</strong>&lt;br&gt;Method to move to turtle to the given x and y location</td>
</tr>
<tr>
<td><code>int</code></td>
<td><strong>getXPos()</strong>&lt;br&gt;Method to get the current x position</td>
</tr>
</tbody>
</table>
Now use your powers

Input: a String to test if inside
Output: something about where the string is

This looks promising
CharSequence?

public final class String
extends Object
implements Serializable, Comparable<String>, CharSequence

boolean contains(CharSequence s)
Returns true if and only if this string contains the specified sequence of char values.

Ok, maybe using the official docs isn’t so easy…
(but it should make more sense later)
JAVA STYLE

• No indent character allowed
• Pressing indent = 2 spaces  
  • (vim and emacs should do this)
• 80 characters per line max
• Good comments
• Google’s Java style guide has much more:
  google.github.io/styleguide/javaguide.html

Sublime and Notepad++ can visually show spaces

“:set list” in vim reveals if there are tabs (tab = ^I)
Good comments

//Write out one line to the file
//Each line is a list of values separated by commas
for (int i = 0; i < values.length; i++) {
    file.write(Double.toString(values[i]).getBytes());
    if (i < values.length - 1)
        file.write(",", .getBytes());
}
file.write("\n").getBytes();

You can get an idea of what it does without really reading the code
Bad comments

```java
for (int i = 0; i < values.length; i++) {
    file.write(Double.toString(values[i]).getBytes());
    if (i < values.length - 1)
        file.write(",",
    .getBytes()); //write comma
}
file.write("\n".getBytes());
```
Homework 2

- `turnToFace()` and `moveTo()` can be super useful
  - Instead of calculating angles, just tell it the exact point to go next
- No magic constants, e.g. `turn(83)`
  - Instead do something like:
    ```
    int w_angle = 83;
    turn(w_angle);
    ```
- Constants often repeat themselves so it’s not so bad
Compiling

javac -cp turtleClasses.jar:. CS11TurtleGraphics.java
java -cp turtleClasses.jar:. CS11TurtleGraphics

- -cp: “classpath”, search for libraries here
- turtleClasses.jar:. search in “turtleClasses.jar” and “.” (current directory)
Questions
Turtle graphics fun