Feb 22: CS 1110: Announcements

- Sign up for a one-on-one!
  - CMS: OPTIONAL: one-on-ones
- Prelim 1 is March 13. You have until March 1st, 11:59pm to register a conflict or a need for accommodation. There is no single makeup session. See website: “Assessment → Exams”
  - CMS: Prelim 1 conflicts
- A2 tentatively: released Wed 2/28, due Wed 3/7

Conditionals: If-Statements

<table>
<thead>
<tr>
<th>Format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>if &lt;boolean-expression&gt;:</td>
<td># is there a new high score?</td>
</tr>
<tr>
<td></td>
<td>if curr_score &gt; high_score:</td>
</tr>
<tr>
<td></td>
<td>high_score = curr_score</td>
</tr>
<tr>
<td></td>
<td>print(&quot;New high score!&quot;)</td>
</tr>
</tbody>
</table>

Execution:
if (Boolean-expression) is true, then execute all of the statements indented directly underneath (until first non-indented statement)

What gets printed? (Question)

```python
a = 0
if a == 0:
    a = a + 1
if a == 0:
    a = a + 2
a = a + 1
print(a)
```

A: 0  B: 1  C: 2  D: 3  E: I do not know

Conditionals: If-Else-Statements

<table>
<thead>
<tr>
<th>Format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>if &lt;boolean-expression&gt;:</td>
<td># who is the winner?</td>
</tr>
<tr>
<td></td>
<td>if score1 &gt; score2:</td>
</tr>
<tr>
<td></td>
<td>winner = &quot;Player 1&quot;</td>
</tr>
<tr>
<td></td>
<td>else:</td>
</tr>
<tr>
<td></td>
<td>winner = &quot;Player 2&quot;</td>
</tr>
</tbody>
</table>

Execution:
if (Boolean-expression) is true, then execute statements indented under if, otherwise execute the statements indented under else

Conditionals: “Control Flow” Statements

<table>
<thead>
<tr>
<th>True</th>
<th>False</th>
</tr>
</thead>
<tbody>
<tr>
<td>if b:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>s1    # statement</td>
</tr>
<tr>
<td></td>
<td>s3    # statement</td>
</tr>
<tr>
<td></td>
<td>False</td>
</tr>
<tr>
<td>if b:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>s1    # statement</td>
</tr>
<tr>
<td></td>
<td>s2    # statement</td>
</tr>
<tr>
<td>else:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>s1</td>
</tr>
<tr>
<td></td>
<td>s2</td>
</tr>
<tr>
<td></td>
<td>s3</td>
</tr>
</tbody>
</table>

Structure vs. Flow

<table>
<thead>
<tr>
<th>Program Structure</th>
<th>Control Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order in which statements are written in scripts and modules</td>
<td>Order in which statements are actually executed at runtime</td>
</tr>
<tr>
<td>Not necessarily the order in which Python executes them</td>
<td>Statements may be:</td>
</tr>
<tr>
<td></td>
<td>• skipped</td>
</tr>
<tr>
<td></td>
<td>• executed more than once</td>
</tr>
</tbody>
</table>

Flow: Program only takes one path each execution

true

false
What does the call frame look like next? (Q)

```
def max(x,y):
    if x > y:
        return x
    return y
max(0,3)
```

Current call frame:

```
A:
    max
    2
    x 0
    y 3
    RETURN 3
```

Program Flow and Variables

Variables created inside `if` continue to exist past `if`:

```
a = 0
if a == 0:
    b = a + 1
    print(b)
```

…but are only created if the program actually executes that line of code

Program Flow and Testing

Can use `print` statements to examine program flow

```
# Put max of x, y in z
print('before if')
if x > y:
    print('inside if x>y')
    z = x
else:
    print('inside else (x<=y)')
    z = y
print('after if')
```

If-Elif-Else (Question)

```
a = 2
if a == 2:
    a = 3
elif a == 3:
    a = 4
print(a)
```

What gets printed?

A: 2  
B: 3  
C: 4  
D: I do not know