Mini-Lecture 11

Conditionals
# Conditionals: If-Statements

## Format

```python
if <boolean-expression>:
    <statement>
    ...
    <statement>
```

## Example

```python
# Put x in z if it is positive
if x > 0:
    z = x
```

## Execution:

if `<boolean-expression>` is true, then execute all of the statements indented directly underneath (until first non-indented statement)
Conditionals: If-Else-Statements

Format

```python
if <boolean-expression>:
    <statement>
    ...
else:
    <statement>
    ...
```

Example

```python
# Put max of x, y in z
if x > y:
    z = x
else:
    z = y
```

Execution:

If `<boolean-expression>` is true, then execute statements indented under `if`; otherwise execute the statements indented under `else`.
Conditionals: “Control Flow” Statements

**if** \( b \):

<table>
<thead>
<tr>
<th></th>
<th>( s1 ) # statement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( s3 )</td>
</tr>
</tbody>
</table>

**else:**

<table>
<thead>
<tr>
<th></th>
<th>( s1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( s2 )</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( s3 )</td>
</tr>
</tbody>
</table>

Branch Point: Evaluate & Choose

Statement: Execute

**Flow**

Program only takes one path each execution

9/19/18

Conditionals
def max(x,y):
    
    """Returns: max of x, y"""
    
    # swap x, y
    # put the larger in y
    if x > y:
        temp = x
        x = y
        y = temp
    return y

• temp is needed for swap
  ▪ x = y loses value of x
  ▪ “Scratch computation”
  ▪ Primary role of local vars

• max(3,0):

<table>
<thead>
<tr>
<th>max</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>0</td>
</tr>
<tr>
<td>y</td>
<td>0</td>
</tr>
<tr>
<td>temp</td>
<td>3</td>
</tr>
</tbody>
</table>
def max(x, y):
    """Returns: max of x, y"""
    # swap x, y
    # put the larger in y
    if x > y:
        temp = x
        x = y
        y = temp
    return y

• temp is needed for swap
  ▪ x = y loses value of x
  ▪ “Scratch computation”
  ▪ Primary role of local vars

• max(3,0):
def max(x, y):
    # swap x, y
    # put the larger in y
    if x > y:
        temp = x
        x = y
        y = temp
    return y

• temp is needed for swap
  ▪ x = y loses value of x
  ▪ “Scratch computation”
  ▪ Primary role of local vars

• max(3, 0):

```
<table>
<thead>
<tr>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>temp</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>RETURN</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>
```

9/19/18
def max(x,y):
    
    """Returns: max of x, y"""
    # swap x, y
    # put the larger in y
    if x > y:
        temp = x
        x = y
        y = temp
    return temp

• Value of max(3,0)?

A: 3
B: 0
C: Error!
D: I do not know
def max(x,y):
    
    '''Returns: max of x, y'''
    
    # swap x, y
    # put the larger in y
    if x > y:
        temp = x
        x = y
        y = temp
    return temp

• Value of max(3,0)?

    A: 3  CORRECT
    B: 0
    C: Error!
    D: I do not know

• Local variables last until
  • They are deleted or
  • End of the function

• Even if defined inside if
**Conditionals and Local Variables**

def max(x,y):
    
    """Returns: max of x, y"""

    # swap x, y
    # put the larger in y
    if x > y:
        temp = x
        x = y
        y = temp

    return temp

• Value of max(0,3)?

    A: 3
    B: 0
    C: Error!
    D: I do not know

9/19/18
def max(x,y):
    """Returns: max of x, y""
    # swap x, y
    # put the larger in y
    if x > y:
        temp = x
        x = y
        y = temp
    return temp

• Value of max(0,3)?
  A: 3  
  B: 0  
  C: Error!  CORRECT  
  D: I do not know

• Variable existence depends on flow
• Understanding flow is important in testing
Conditionals: If-Elif-Else-Statements

### Format

```python
if <boolean-expression>:
    <statement>
    ...
elif <boolean-expression>:
    <statement>
    ...
else:
    <statement>
    ...
```

### Example

```python
# Put max of x, y, z in w
if x > y and x > z:
    w = x
elif y > z:
    w = y
else:
    w = z
```
## Conditionals: If-Elif-Else-Statements

### Format

<table>
<thead>
<tr>
<th>if &lt;boolean-expression&gt;:</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;statement&gt;</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>elif &lt;boolean-expression&gt;:</td>
</tr>
<tr>
<td>&lt;statement&gt;</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

... else:

|    <statement>         |
|                         |
| ...                    |

### Notes on Use

- No limit on number of 
  - Can have as many as want
  - Must be between if, else
- The else is always optional
  - if-elif by itself is fine
- Booleans checked in order
  - Once it finds a true one, it skips over all the others
  - else means all are false
Conditional Expressions

Format

\[ e_1 \textbf{if} \ bexp \ \textbf{else} \ e_2 \]

- \( e_1 \) and \( e_2 \) are any expression
- \( bexp \) is a boolean expression
- This is an expression!

Example

# Put max of x, y in z
\[
z = x \textbf{if} \ x > y \ \textbf{else} \ y
\]

expression, not statement