

## Mini-Lecture 11

# Conditionals

# Conditionals: If-Statements

---

## Format

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

## Example

```
# 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

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

## Example

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

### Execution:

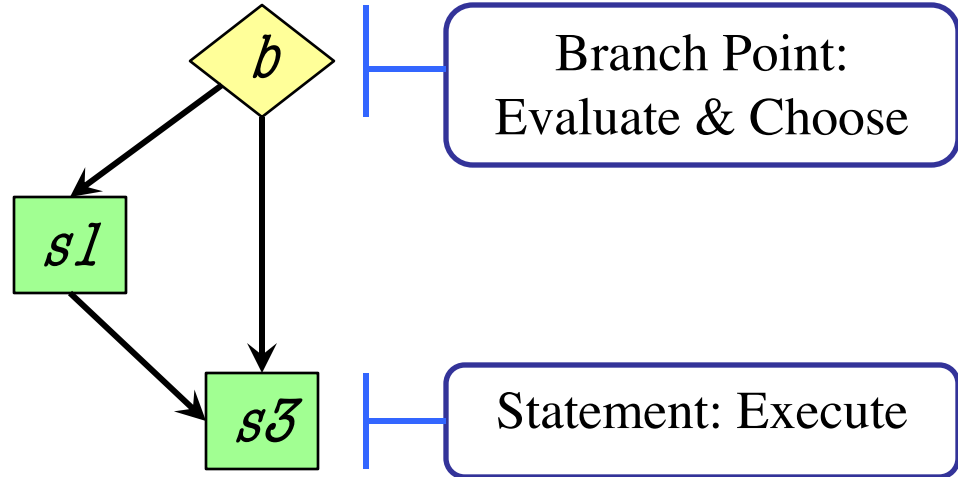
if <*boolean-expression*> is true, then execute statements indented under if; otherwise execute the statements indented under elsec

# Conditionals: “Control Flow” Statements

**if**  $b$ :

|  $s1$  # statement

$s3$



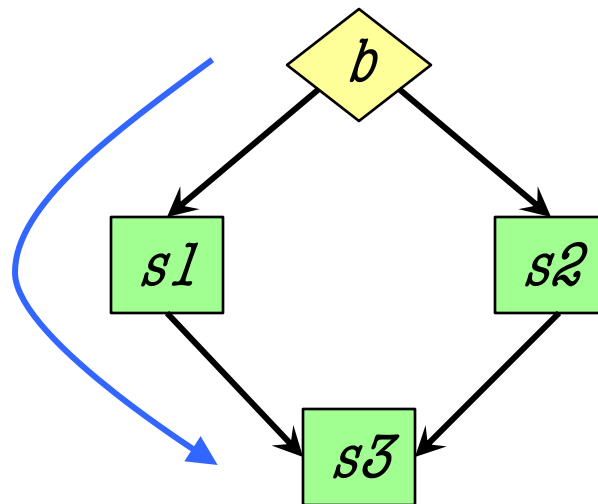
**if**  $b$ :

|  $s1$

**else:**

|  $s2$

$s3$



**Flow**

Program only takes one path each execution

# Conditionals and Local Variables

```
def max(x,y):
```

```
    """Returns: max of x, y"""
```

```
    # swap x, y
```

```
    # put the larger in y
```

```
1  if x > y:
```

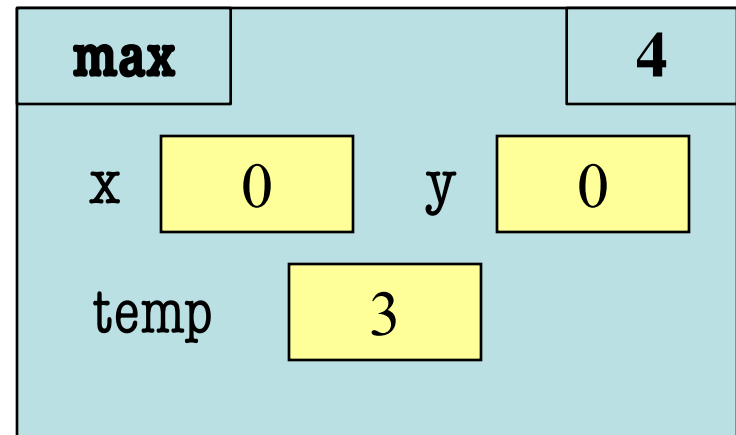
```
2  |     temp = x
```

```
3  |     x = y
```

```
4  |     y = temp
```

```
5  return y
```

- temp is needed for swap
  - x = y loses value of x
  - “Scratch computation”
  - Primary role of local vars
- max(3,0):



# Conditionals and Local Variables

```
def max(x,y):
```

```
    """Returns: max of x, y"""
```

```
    # swap x, y
```

```
    # put the larger in y
```

```
1  if x > y:
```

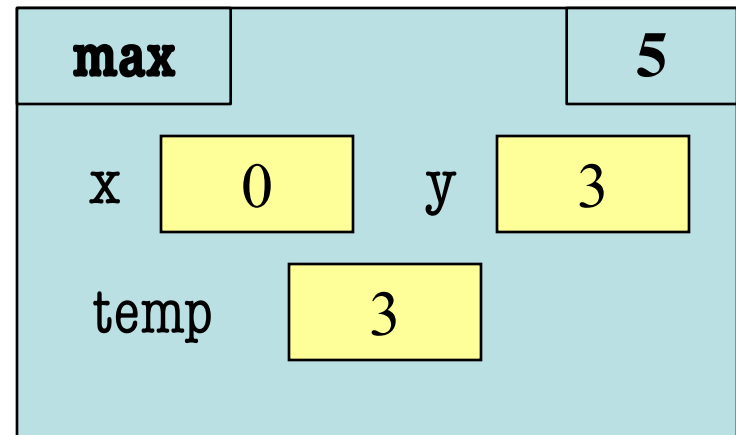
```
2  |     temp = x
```

```
3  |     x = y
```

```
4  |     y = temp
```

```
5  return y
```

- temp is needed for swap
  - x = y loses value of x
  - “Scratch computation”
  - Primary role of local vars
- max(3,0):



# Conditionals and Local Variables

```
def max(x,y):
```

```
    """Returns: max of x, y"""
```

```
    # swap x, y
```

```
    # put the larger in y
```

```
1  if x > y:
```

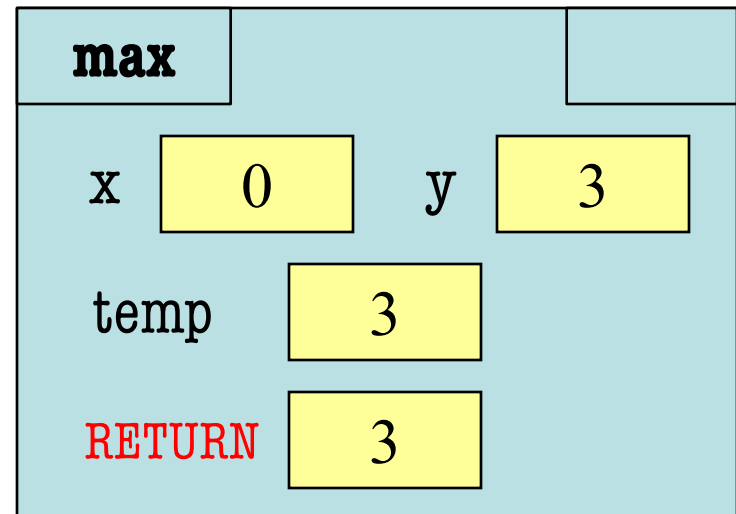
```
2  |     temp = x
```

```
3  |     x = y
```

```
4  |     y = temp
```

```
5  return y
```

- temp is needed for swap
  - x = y loses value of x
  - “Scratch computation”
  - Primary role of local vars
- max(3,0):



# 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(3,0)?

A: 3

B: 0

C: **Error!**

D: I do not know



# 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(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

# 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!** **CORRECT**

D: I do not know

- Variable existence depends on flow
- Understanding flow is important in testing

# Conditionals: If-Elif-Else-Statements

---

## Format

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

## Example

```
# 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

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

## Notes on Use

- No limit on number of elif
  - 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

---

$e_1$  **if**  $b_{exp}$  **else**  $e_2$

- $e_1$  and  $e_2$  are any expression
- $b_{exp}$  is a boolean expression
- This is an expression!

## Example

---

# Put max of  $x$ ,  $y$  in  $z$

$z = x$  **if**  $x > y$  **else**  $y$



expression,  
not statement