

## 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

### Format

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

### Example

```
# is there a new high score?
if curr_score > high_score:
    high_score = curr_score
    print("New high score!")
```

### Execution:

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

## What gets printed? (Question)

```
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

### Format

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

### Example

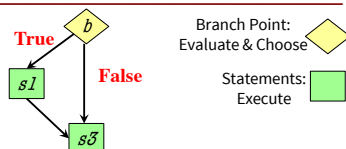
```
# who is the winner?
if score1 > score2:
    winner = "Player 1"
else:
    winner = "Player 2"
```

### Execution:

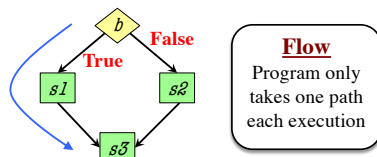
if (*Boolean-expression*) is true, then execute statements indented under **if**; otherwise execute the statements indented under **else**

## Conditionals: "Control Flow" Statements

```
if b:
    s1 # statement
    s2 # statement
```



```
if b:
    s1
else:
    s2
    s3
```



## Structure vs. Flow

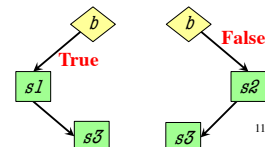
### Program Structure

- Order in which statements are **written** in scripts and modules
- Not necessarily the order in which Python executes them

```
if b:
    s1
else:
    s2
    s3
```

### Control Flow

- Order in which statements are **actually executed** at runtime
  - Statements may be:
    - skipped
    - executed more than once



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

```
def max(x,y):
1 | if x > y:
2 | | return x
3 | return y
```

max(0,3)

Current call frame:

max	1
x	0
y	3

A:

max	2
x	0
y	3

B:

max	
x	0
y	3
RETURN	0

C:

max	
x	0
y	3
RETURN	3

D:

max	3
x	0
y	3

17

## 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

22

## Control Flow and Variables (Q1)

```
def max(x,y):
    """Returns: max of x, y"""
    # note: code has a bug!
    # check if x is larger
    if x > y:
        bigger = x
    return bigger
```

maximum = max(3,0)

Value of maximum?

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

25

## Control Flow and Variables (Q2)

```
def max(x,y):
    """Returns: max of x, y"""
    # note: code has a bug!
    # check if x is larger
    if x > y:
        bigger = x
    return bigger
```

maximum = max(0,3)

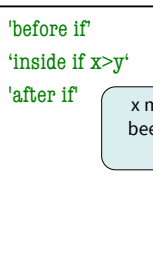
Value of maximum?

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

27

## Program Flow and Testing

Can use **print** statements to examine program flow



x must have been greater than y

```
# 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')
```

29

## 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

32