

# CS21: INTRODUCTION TO COMPUTER SCIENCE

---

Prof. Mathieson

Fall 2018

Swarthmore College

# Outline Sept 19:

Sit at a new computer!

- Relational operators (<, >, ==, etc)
- Conditionals (if/elif/else)
- Logical operators (and, or, not)
- Practice: favorite number programs
- Friday after quiz: accumulator + conditionals!

## Notes

- **Lab 2** due **Saturday** night
- **Quiz 1**: this Friday! (9/21), first 25-30 min (review study guide)
- **Practice problems** in the practice directory
- **Ninja session tonight! 7-10pm** (this room)

# Study strategies

- $10 - (2.5 \text{ class} + 1.5 \text{ lab}) = \mathbf{6 \text{ hrs/week outside of class}}$
- Go over class notes
- Finish in class problems and then go over solutions
- Complete practice problems
- Quiz study guide problems
- Start the lab before your lab section
- Work on lab problems after lab
- Ninja sessions, office hours, meet with instructors

"""

Write a program that asks the user for their first and last name (assume entered in lowercase) and returns their Swarthmore username. For example:

```
python3 username.py
```

```
Enter your first name: barack
```

```
Enter your last name: obama
```

```
Username: bobama1
```

```
Email: bobama1@swarthmore.edu
```

```
Author: Sara Mathieson
```

```
Date: 9/17/18
```

```
"""
```

```
def main():
```

```
    # ask for user input
```

```
    first = input("Enter your first name: ")
```

```
    last = input("Enter your last name: ")
```

```
    number = int(input("Enter your number: "))
```

```
    num_chars = len(last)
```

```
    # initialize accumulator to first character of first name
```

```
    username = first[0]
```

```
    # use the min(a,b) function to see if we need 6 chars, or fewer
```

```
    num_iter = min(6, num_chars)
```

```
    for i in range(num_iter):
```

```
        # accumulate characters of the last name
```

```
        username = username + last[i]
```

```
    username = username + str(number) # add on user's number
```

```
    print("Username:", username)
```

```
    print("Email:", username + "@swarthmore.edu")
```

```
main()
```

# Booleans and comparison operators

# New type: booleans

- Can only be True or False
- The result of a logical expression
- Comparison or relational operators

- Less than: <
- Greater than: >
- Less than or equal to: <=
- Greater than or equal to: >=
- Is equal to: ==
- Is not equal to: !=

```
>>> 3 < 5
True
>>> 7 > 10
False
>>>
>>> 3.0 == 3
True
>>>
>>> x = "hello"
>>> x != "hello "
True
>>>
>>> x == "hello"
True
```

# New control statements: if/elif/else

- Idea: if something is thing is **true**, we want one thing to happen
- If something is **false**, we want another thing to happen
- Example:

**If it is raining:**

I will wear rain boots

**If it is not raining:**

I will wear sandals

# New control statements: if/elif/else

- Control statements:

- 1) Functions (keyword: **def**, then indent afterwards)
- 2) For-loops (keyword: **for**, then indent afterwards)
- 3) If-statements (keyword: **if**, then indent afterwards)

- If-statement syntax:

- **if <condition>:**  
    <statements> # Executed if <condition> is True



# New control statements: if/elif/else

Example: based on class year, has a student graduated or not?

**if <condition1>:**

**<statements\_1> # Executed if <condition1> is True**

**elif <condition2>:**

**<statements\_2> # Executed if <condition1> is False and  
<condition2> is True**

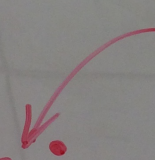
**else:**

**<statements\_3> # Executed if <condition1> and <condition2>  
are both False**

"GETS"

← ASSIGNMENT

= ← EQUALS

else:  NO CONDITIONALS  
↑  
COLONS!

# Try it out!

- conditionals1.py

```
python3 conditionals1.py  
Enter your favorite integer: -4  
Your number is negative!
```

- conditionals2.py

```
python3 conditionals2.py  
Enter your favorite integer: -4  
Your number is even!
```

- movie.py

# Which of these are valid?

1)

```
if  
elif  
elif  
elif  
else
```

2)

```
else  
if
```

5)

```
if  
else  
if
```

3)

```
elif  
else
```

4)

```
if  
if  
if  
if
```


# Which of these are valid?

1)

```
if  
elif  
elif  
elif  
else
```

2)

```
else  
if
```



5)


```
if  
else  
if
```

4)

```
if  
if  
if  
if
```

3)

```
elif  
else
```



# Logical operators



if <sup>A</sup>not sunny:  
library

A	not A
F	T
T	F

## Logical operators (and, or, not)

if <sup>A</sup>sunny and <sup>B</sup>finish\_HW:  
beach # not valid

if sunny or finish\_HW:  
beach

inclusive or

truth table

A	B	A and B
F	F	F
F	T	F
T	F	F
T	T	T

A	B	A or B
F	F	F
F	T	T
T	F	T
T	T	T

# Try it out!

- movie.py

```
"""
```

```
Assume that movie tickets have the following prices based on age:
```

```
    0 – 12 years: $8
```

```
    13 – 64 years: $12
```

```
    65 and up: $8
```

```
Write a program to compute the ticket price based on the user's age. Try:
```

```
1) a method using if/elif/else
```

```
2) a method using only if/else, along with or
```

```
3) a method using only if/else, along with and
```

```
4) a method using only if/else!
```

```
Author:
```

```
Date:
```

```
"""
```



# New practice problems (suggested order)

- **voting.py** conditionals
- **quadratic.py** conditionals
- **coin\_toss.py** for loops, accumulator pattern, nesting, conditionals, random library
- **vowels.py** for loops, accumulator pattern, nesting, conditionals, indexing