On your way in…(on the side table)

Pick-up:

1. POGIL Activity #8
2. POGIL Activity #9 (and 10, combined)

Hand-in your homework

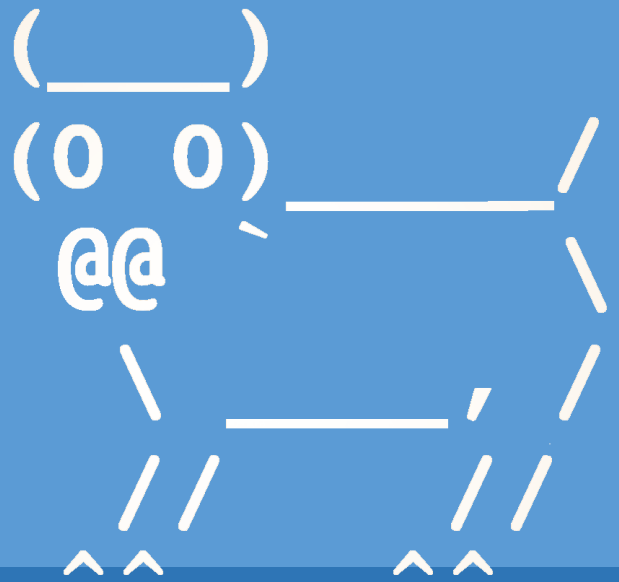
(2 piles: ID under 45 or over 45)

Welcome to CS 134!

Introduction to Computer Science

Iris Howley

-Abstraction-



Grace Hopper/Richard Tapia Celebration of Women/Diversity in Computing Info Session Tonight

- Monday February 11 at 9:15pm in the Computer Science Lounge (outside my office, TCL 308)
- Interested in learning more or attending either of these? Come to the info session!
- We'll have students who have attended previously!

Have you found the TA + Office Hours?

Office Hours:	Iris (TCL308): Tues. TBD, Wed. 12:30-2:30p, Thurs. 1-2:30p Duane (TPL3[illegible]): Mon. 2:30-4:30p, Tues. 7:30-9p, Thurs. 9:45-11:20a
Lectures:	MWF 11-11:50 Chemistry 125 (Wege)
Labs:	M 1-2:30pm, 2:30-4pm, T 10-11:30am, 1-2:30pm, 2:30-4pm, TCL 217a
Textbook:	*Think Python (2nd Edition)*, found at greentreepress.com and here
TAs:	Noah Andrew, Chris Anton, Will Burford, Jimmy DeLano, Jacob Justh, Julia Kawano, Aidan Lloyd-Tucker, Grace Mazzarella, [illegible]
TA Hours:	Sun. 4-10pm, Mon-Thu. 7-10pm, +Wed. 2-4p (in TCL312), +Wed. 4:30-6:30pm, Thu. 6-10p. (All office hours in TCL217a, unless otherwise described)

Office hours for TAs & Professors have begun!

If Statements & Booleans

- `return x//2 == 0`

Is equivalent to (and much better than):

- `if x//2 == 0:`
  - `return True`
- `else:`
  - `return False`

One-line Python Conditionals

- `x*3+1 if odd(x) else x//2`

Is equivalent to (but not always better than):

- `if odd(x):`
  - `x*3+1`
- `else:`
  - `x//2`

Readability is often more important than conciseness

While Loops

```
def odd(x):
    """Return True iff x is an odd integer."""
    return x%2 == 1

def syr(x):
    """Compute the 3n+1 function value associated with x.
    >>> syr(10)
    5
    >>> syr(3)
    10
    """
    return x*3+1 if odd(x) else x//2

def orbit(x):
    """Print the orbit of syr on x: apply syr to x until it becomes 1."""
    while x != 1:
        print(x)
        x = syr(x)
    print(x)


# This line is for running as a script
orbit(int(input("Number to start with: ")))
```

For Loops

What does #1 do?

#1

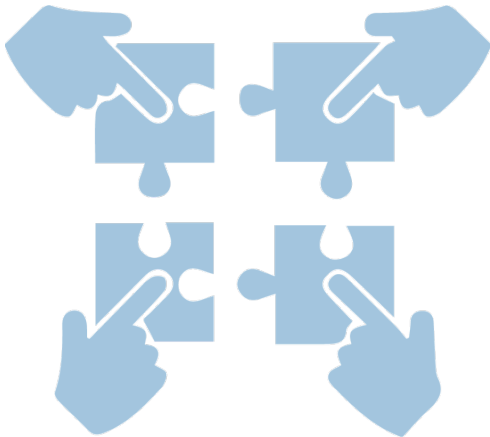
```
name = input("Enter name:")
x = 0
while (x<20):
      print(name)
      x = x + 1
```

#2

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

for x in range(20):
      print(name)
```

What does #2 do?



Iris
Iris
Iris
Iris
Iris
Iris
Iris
Iris
Iris
Iris
Iris
Iris
Iris
Iris
Iris
Iris
Iris
Iris
Iris
Iris
Iris
Iris

For Loops

#1

```
name = input("Enter name:")
x = 0
while (x<20):
    print(name)
    x = x + 1
```

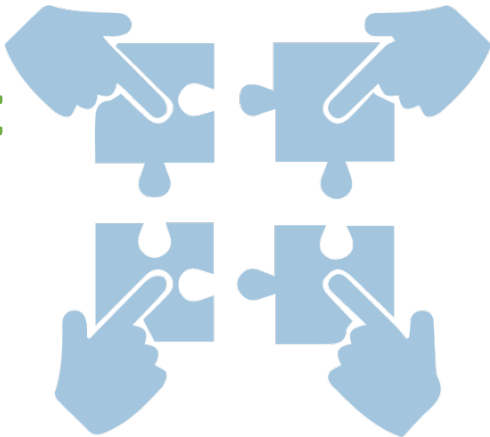
…will print "Iris" a total of 20x on 20 lines

#2

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

for x in range(20):
    print(name)
```

…will ALSO print "Iris" a total of 20x on 20 lines



For Loops

What does #3 do?

#3

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

for letter in name:
    print(letter + "*")
```

```
i*
r*
i*
s*
```

#2

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

for x in range(20):
    print(name)
```

TODAY'S LESSON

Abstraction makes programming GREAT

(abstraction, encapsulation, etc.)

Classes, Objects

Everything in Python is an object!

The textbook has really great activities to step through, with exercises to do at the end.

Chapter 4: Case study: interface design

The Syracuse Function

Running as a Script

```
        x = syr(x)
    print(x)


# This line is for running as a script
orbit(int(input("Number to start with: ")))
```

A

In interactive python, (A) will *always* ask the user for a `Number to start with`!

With (B), that line only happens if the code is being run as a script, not if you want to use it as a library in interactive python!

```
        x = syr(x)
    print(x)


if __name__ == "__main__":
    orbit(int(input("Number to start with: ")))
```

B

Testing Comments

```
def syr(x):
    """Compute the 3n+1 function value associated with x.
    >>> syr(10)
    5
    >>> syr(3)
    10
    """
    return x*3+1 if odd(x) else x//2
```

What are these comments for?

The line that starts with '>>>' is a sample function call for this function
The immediately following line is the anticipated output

i.e., if we call the syr() function with a value of 10, we expect this function to return the number 5.

Testing Comments

```
def syr(x):
    """Compute the 3n+1 function value associated with x.
    >>> syr(10)
    5
    >>> syr(3)
    10
    """
    return x*3+1 if odd(x) else x//2
```

What are these comments for?

If you place `import doctest` `doctest.testmod()` under `if __name__ = "__main__"`

This will run these tests and make sure your code is returning the values you expect (i.e., helpful for testing the logic of your program)

Format Printing

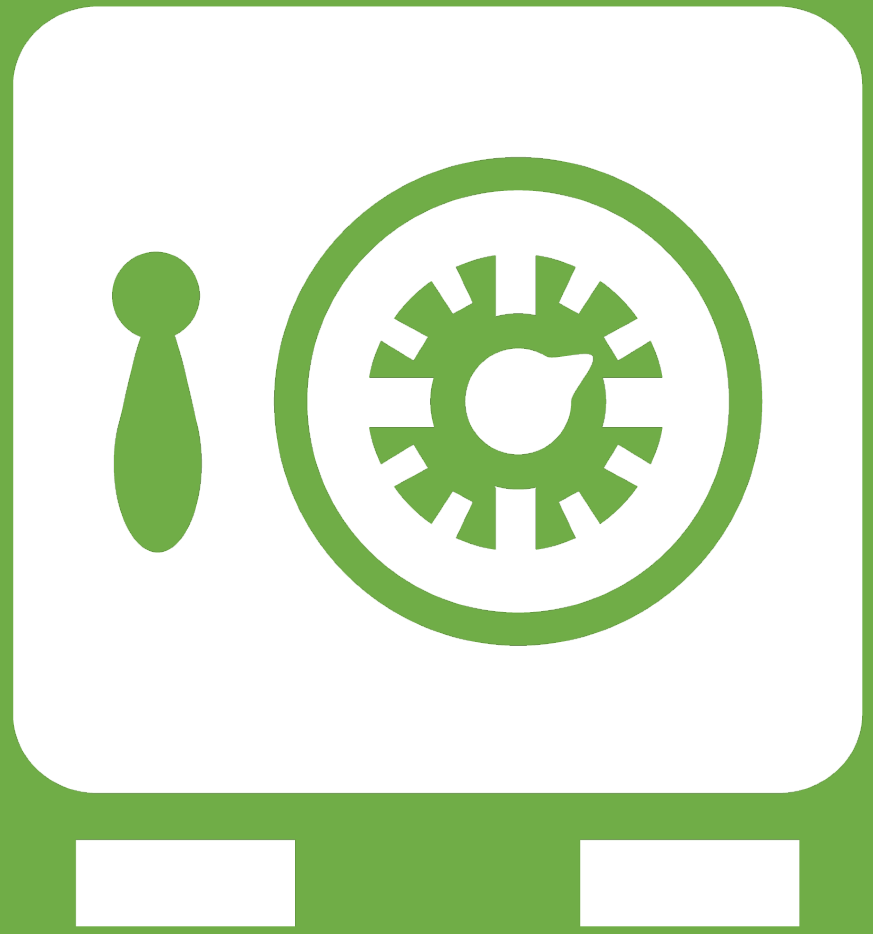
```
print(“{} was born on {}/{}/{}".format(“Pixel”,5,16, 2018))
```

Pixel was born on 5/16/2018

This will print the same exact text:

```
name = “Pixel”
month = 5
day = 16
year = 2018
print(“{} was born on {}/{}/{}".format(name,month,day, year))
```


QUESTIONS?



Leftover Slides

Everything in Python is an Object

- Even functions!

```
def do_something():
   return 'hello world'
def run_this_func(new_func):
   result = new_func()
   return result

run_this_func(do_something)
```

Syracuse Function

1. Start with any positive integer *n*
2. The next term is determined by n:
   - If n is odd, the next term is 3*n + 1
   - If n is even, the next term is n/2

Collatz (1937): “no matter what value of *n*, the sequence will always reach 1”

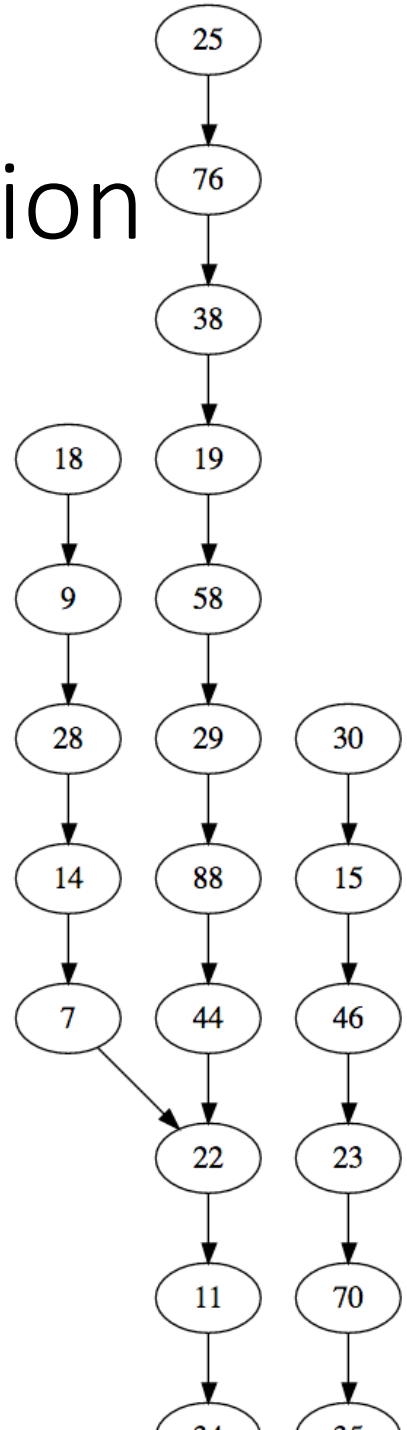
Erdős: "Mathematics may not be ready for such problems."

Lagarias (2010): "this is an extraordinarily difficult problem, completely out of reach of present day mathematics."

Syracuse Function

- The orbit for 4:
  - 4 → 2 → 1
- The orbit for 5:
  - 5 → 16 → 8 → 4 → …

Syracuse Function



https://upload.wikimedia.org/wikipedia/commons/a/ad/Collatz-graph-all-30-no27.svg

How to program the Syracuse Function?

1. Start with any positive integer *n*
2. The next term is determined by n:
   - If n is odd, the next term is 3*n + 1
   - If n is even, the next term is n/2