Lecture 20:
Programming with Subclasses

CS 1110

## Introduction to Computing Using Python
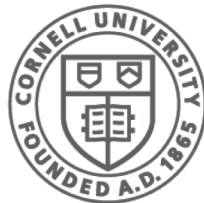
[E. Andersen, A. Bracy, D. Gries, L. Lee, S. Marschner, C. Van Loan, W. White]

Put Me in the Zoo

- Classes: **Animal, Bird, Fish, Penguin, Parrot**
- Instances can **swim**, **fly**, and **speak** based on class membership
- Track:
  - \# of animals created (Q1)
  - **name**, **tag #**, **weight** for each animal (w/default weights)
- Methods:
  - speak(words): print words if animal speaks (Q2)
  - eat(): print eating sounds & gain 1 pound (Q3)

Questions to ask

- What does the class hierarchy look like?
- What are class attributes? What are instance attributes? What are constants?
- What does the __init__ function look like?
- How do we support default weights?
- How do we implement the class methods?
- What does a "*stringified*" Animal look like? str(a)
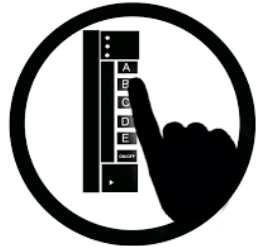
Q1: What is the best way to keep track of the number of Animals that have been created?

A: a global variable that you increment each time you call the Animal constructor

B: a class attribute inside the Animal Class that is incremented by the Animal's __init__ method

C: an instance attribute inside each Animal that is incremented by the Animal's __init__ method

D: A & B both work, but B is better
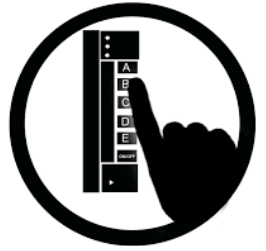
E: A & B & C all work, but C is best
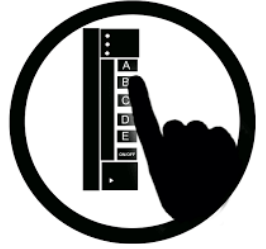
If speak is defined by the Animal Class like this:

```
def speak(self, words):
    if self.CAN_SPEAK:
        print(words)
```

Q2: Which subclasses need to provide their own version of this method?

A: Bird, Fish, Penguin, and Parrot
B: Bird and Parrot
C: just Parrot
D: none
E: I don't know

If eat is defined by the Animal Class like this:

```python
def eat(self):
    print("NOM NOM NOM")
    self.weight += 1
```

Q3: We want Fish to say nothing and Birds to make a pecking sound. Which subclasses need to provide their own version of this method?

A: Bird, Fish, Penguin, and Parrot
B: Bird and Fish
C: just Bird
D: just Fish
E: I don't know