Announcements

- Homework
  - I've posted HW9
  - Both 8 and 9 are due on Friday
  - I finished grading 6 yesterday and pushed results back to your Github
- From what I've seen, I'm really excited to read over and score your midterm projects, but it will probably take me a week or so
- I'm planning to get grade reports given out for what I do have graded later today or tomorrow
- Polling: `rembold-class.ddns.net`

Review Question

```python
class Demo:
    def __init__(self):
        self.x = []
    def add(self,v):
        self.x.append(v)
    def get_x(self):
        return self.x

A,B = Demo(), Demo()
A.add(3)
B.add(3)
C = B.get_x()
C.append(8)
print(A.get_x() == B.get_x())
```

The code to the left defines a new class, creates some instances of that class and then manipulates them a bit. What is printed to the screen in the last line?

A) True

B) False

C) None, as Demo has no __eq__ method

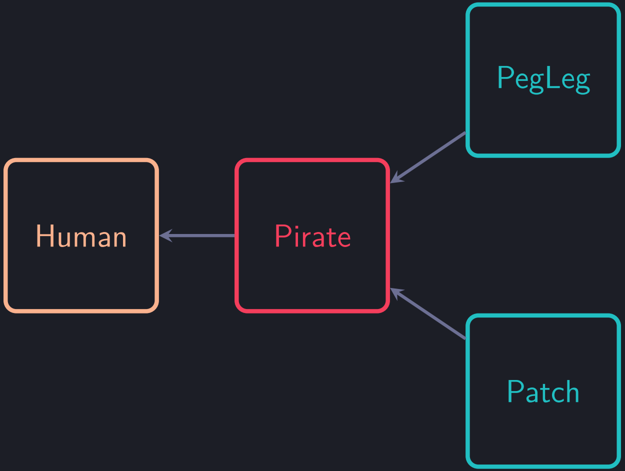
D) This code would error out before completion

Solution: False

Reminders

- We specify what a class's parent is immediately following the class's name
  - Surround parent's name is parentheses
  - `class Human(Animal):`
  - The default parent is type `object`
- All data attributes and methods defined for the parent are then defined for the child as well
  - We are then free to:
    - Add more methods
    - Add more attributes
    - Change inherited methods or attributes

A Swashbuckling Example



Adding or Changing Data Attributes

- The `__init__` method is inherited from the parent class like all other methods!
- If we want other or more stuff to be initialized, we need to redefine `__init__`.
- Have some options in terms of how to do so:
  - Redefine all the inherited attributes and then add our new attributes
  - Call the parent `__init__` method and then add our own beneath

A Change is Coming

- If redefining, ensure that all parent attributes are still present in the child!
  - If you create a child object and plug it in somewhere in place of a parent object, it should still work!
- To call the parent `__init__` method you need to access it through the `ClassName.__init__` structure
  - Example:

```
class Pirate(Human):
    def __init__(self, age): #redefining __init__
        Human.__init__(self, age)    #calling parents __in
```

Understanding Check

What expression would best fill in the gap in the code to the right?

A) `self.wage = wage`

B) `TechJob.__init__(self, wage)`

C) `TechJob.__init__(wage)`

D) `Job.__init__(wage)`

```
class Job(object):
    def __init__(self, wage):
        self.wage = wage

class TechJob(Job):
    def __init__(self, wage):
        self.wage = wage
        self.code = True

class SeniorDev(TechJob):
    def __init__(self, wage, exp):
        # What goes here?
        self.exp = exp
```

Solution: `TechJob.__init__(self, wage)`