CS 150: Classes and Objects

Cynthia Taylor
Oberlin College
March 17th 2014

Midterm Evaluation

- Please take 5 minutes to fill out some feedback on the course

- You do not need to put your name

Clicker Group Change

- I will reshuffle groups over spring break
- If you want to stay with your group, one of you needs to fill out form on blackboard by FRIDAY
- If you do nothing, your group will change

Students

- We want to store students' names, their year, and their most recent three grades

- Could use a nested list

```
students = [['Cynthia', 4, ['A', 'A+', 'A']]
            ['Bob', 2, ['B+','A-','B']]]
```

Problems

- We are imposing unnecessary order on the elements (why is year second and grades third?)
- The elements do not have names
- No convenient way to print students
- Not easy to support multiple types of students (e.g. first-year students may have no prior course grades)

student[c].grade

More problems

- Also no way to ask a student to “do something” (like calculate average grade)

Classes

- Sometimes, the built-in objects are not natural for supporting new types of data
- By defining a new class, we can add a new type to Python
- We can then make *objects* of that type (e.g. objects of type Student)

Methods

- Methods are actions specific to each type of object
  - e.g. for spaceship: move, turn, shoot, dodge, raise shields, lower shields, teleport

- Function associated with an object

Attribute

- An attribute is a feature or characteristic of an object student - grade, year, name
- Unlike a method, it is not an action
- An attribute is something that an object has, not something the object does

Example

- Class name: `SpaceShip`

- Possible methods: scan object
  takeoff, land, synthesize food
- Possible attributes:
  type of food, speed, color of capts chair
  type of weapons, # of weapons, power of force field

Which of the following is not a possible method for a `Car` class?

A. `open_window`

B. `accelerate`

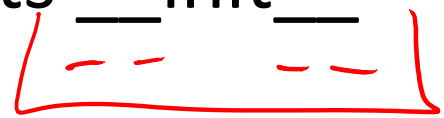
C. `num_wheels`

D. `turn_right`

E. I don't know

Creating Objects

- When an object is created, its __init__ method is called



- __init__ is known as a constructor because it constructs objects

- Inside __init__, assign values to the object's attributes

Example: Point Class

```
class Point:

  def __init__ (self):
     self.x = 0
     self.y = 0
```

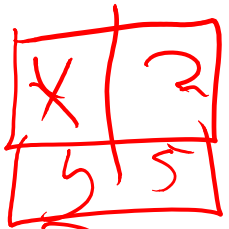
Example: Point Class

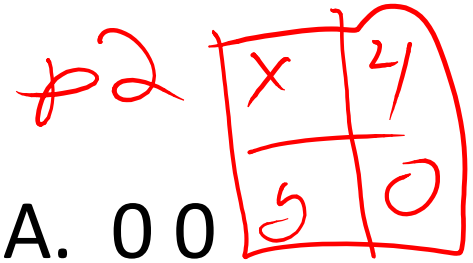
- Don't forget the `self.` before each attribute
- In all methods you write, `self` means "current object"
- You can create a point using `p = Point()`
- `p.x` and `p.y` access the attributes of p

What is the output of this code?

```
p1 = Point()
p1.x = p1.x + 2
p1.y = p1.x + 3
p2 = Point()
p2.x = p2.x + 4

print(p2.x, p2.y)
```





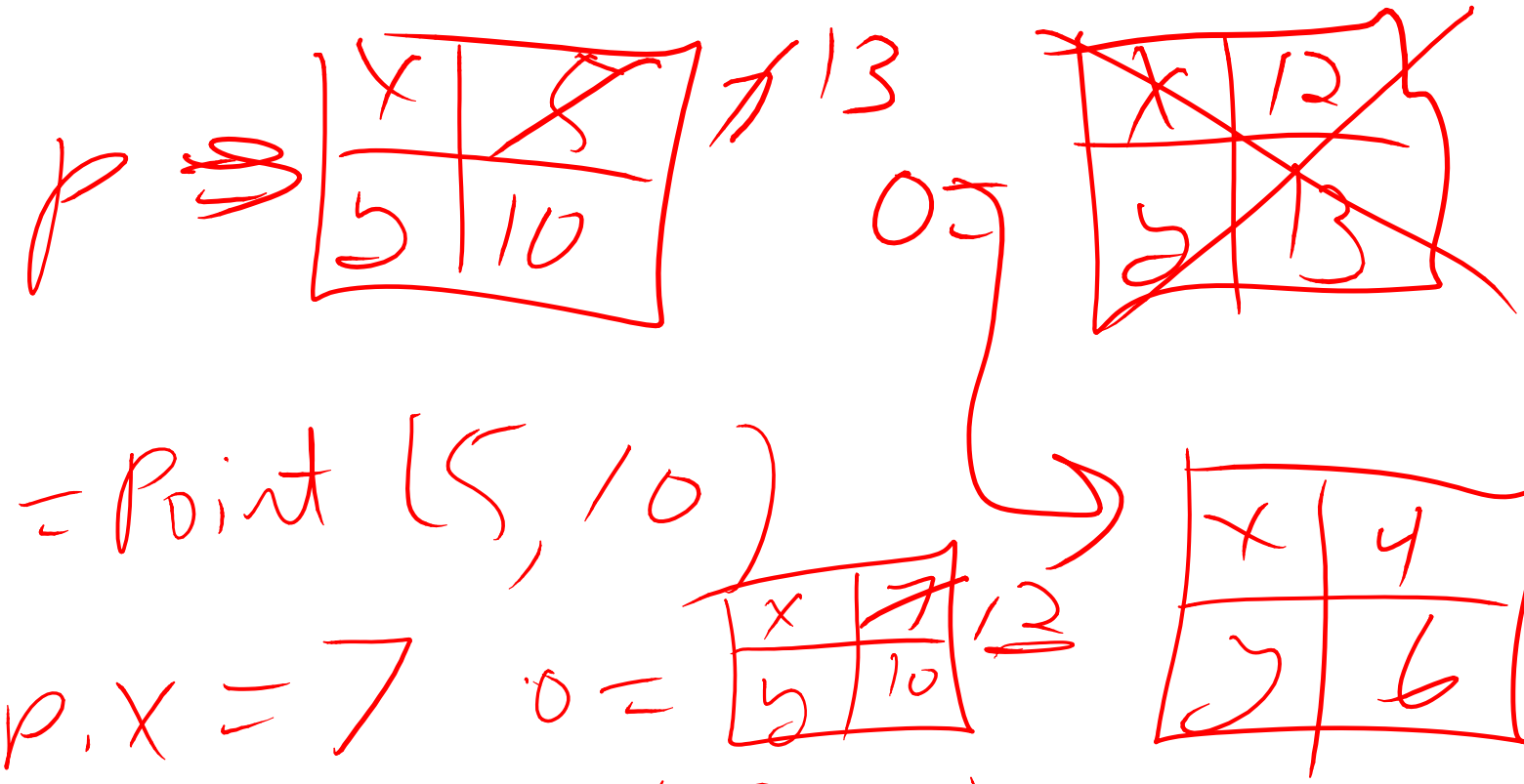
A. 0 0
B. 4 0
C. 2 3
D. 7 3
E. I don't know

Creating Objects with Parameters

- A Point that always starts at (0, 0) may not be as useful as a Point that can start at a specified location.

```
class Point:
    def __init__ (self, x, y):
        self.x = x
        self.y = y
```

- Now create a Point using `p = Point(2, 5)`
- `p.x` and `p.y` then have these initial values



p = Point (5, 10)

p.x = 7

o = Point (12, 13)

o = Point (4, 6)

o.x = 12

o = Point (p.x, p.y)

p = Point(5, 10)

o = p

o.x = 12

Next Class

- Classes and Objects
  - Read Sections 10.1 – 10.2
- Lab 6 – Due Tuesday at 10 pm
- NO PRELAB