Implementing Classes – Key Concepts

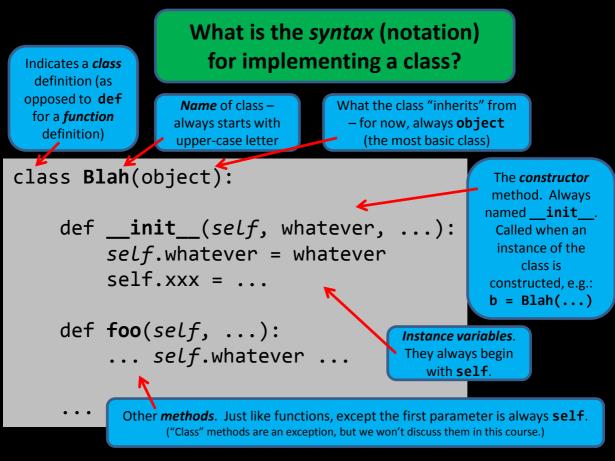
Why are Classes useful?

- A class provides a way to **bundle/organize** data and actions under a single name. Data is stored in *instance variables* and actions are specified by *methods*.
- A class specifies a *type* of thing, from which we can produce multiple *instances*. All instances have the same form (i.e., same names for instance variables and methods), but each has its own data.

As such, a class is like *int* or *float*, but the *programmer* gets to specify the type of data and what can be done with it.

Functions are important in part because we can call them with different arguments to get different results. Classes extend that idea to include *objects with different data*, as well as methods that can be called with different arguments.

• Designing solutions to problems by thinking about the *types* of things needed for the solution, and the *relationships* between those types of things, is fundamental to **object-oriented design**.



Why is the __init__ method special?

class Point(object):
 def __init__(self, x, y):
 self.x = x
 self.y = y
 self.number_of_moves = 0

Causes the Point class' ___init___ method to be called, with:

x = 100 y = 50 self = the Point being constructed

Then sets **p1** to that Point.

Causes the Point class' __init__ method to be called, with: x = 77 y = 33 self = the Point being constructed Then sets p2 to that Point.

def main():
 p1 = Point(100, 50)
 p2 = Point(77, 33)

class Point(object):

def move_by(self, dx, dy):
 self.x = self.x + dx
 self.y = self.y + dy

It is the *same p1* all 3 times. So the one and only **p1** is set to (100, 50), then moves to (120, 80), then moves to (121, 81) in this code snippet.

What is *self*?

Answer: It is set to the *thing before the dot* when a method of the class is called.

def main():

p1 = Point(100, 50)
p2 = Point(77, 33)
p1.move_by(20, 30)
p2.move_by(11, 22)
p1.move_by(1, 1)

self = p2 in the
other two statements.
So the one and only p2
is set to (77, 33) and
moves to (88, 55).

What is *self*. blah? Answer: It is the *blah* instance variable for the *thing before the dot*.

