An Example Python Class

class Account(object):
    def __init__(self, name, balance):
        self.name = name
        self.balance = balance

    def deposit(self, amt):
        self.balance += amt

    def withdraw(self, amt):
        self.balance -= amt

    def inquiry(self):
        return self.balance

The Details

- **object** is the root of the class tree
- use it as superclass when you are not extending another class
- **__init__** implicitly defines instance variables
- **self** is a parameter to all instance methods
- but you don’t include it in the parameter list when you call an instance method
- technically you can use any identifier in place of self (but don’t do it!)

Augmenting the Example:

class Account(object):
    num_accounts = 0
    def __init__(self, name, balance):
        self.name = name
        self.balance = balance
        Account.num_accounts += 1
    def __del__(self):
        Account.num_accounts -= 1

    def deposit(self, amt):
        self.balance += amt

    def withdraw(self, amt):
        self.balance -= amt
More Details

- Class variables are like static variables in Java
- They belong to the class, not to an instance
- All instances share a single copy of a class variable
- __del__ is often absent
  - It is used to do things such as update class variables (as in our example), close network connections, release locks, etc.
- Calling del on an object does not necessarily invoke __del__ -- del reduces reference count

Creating Instances:

```python
from account import Account

checking = Account('David', 50000)
savings = Account('David', 1000000)

# It's payday
checking.deposit(25000)

# Can I buy a 488?
if savings.inquiry() > 500000:
    print("Go shopping. Make sure it's red.")

# Prints 2
print('Created: {}'.format(Account.num_accounts))
```