Py ch12: Object-Oriented Programming

- *Announcements*

30 Oct 2006
CMPT14x
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Review from last time (§9.6-9.13)

- User-defined types: Records
  - Records in M2
- Using Python classes to make records
- Objects are instances of classes, created by constructor functions
- Objects are mutable: copy vs alias
- Using 'id' to check for aliases

What's on for today (Py ch12)

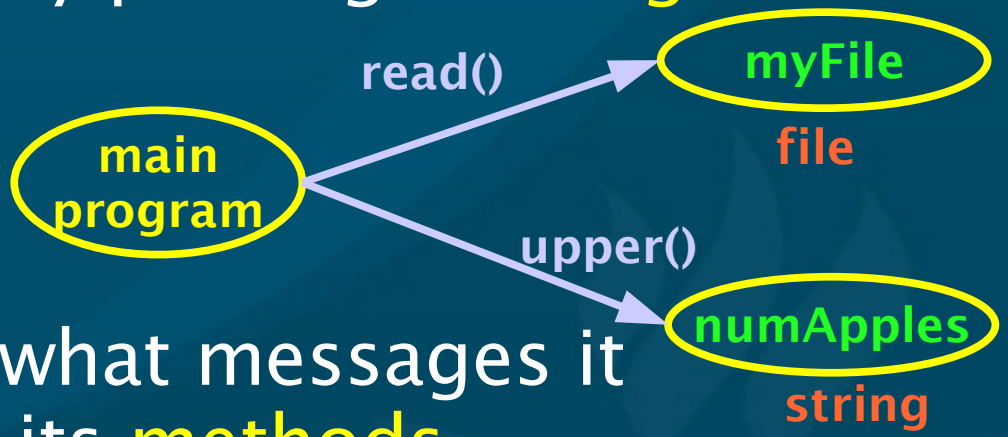
- Object-oriented programming paradigm
- Objects, methods, attributes
- Classes, instances
- Alias vs. shallow copy vs. deep copy

Object-oriented programming

- Procedural paradigm: programs as lists of actions
  - Focus is on the procedures (verbs)
  - Variables, data structures get passed into procedures
    - e.g.: `string.upper('hello')`
- Object-oriented paradigm: collections of objects
  - Focus is on the data (nouns)
  - Messages get passed between objects
  - Procedures are methods belonging to objects
    - e.g.: `'hello'.upper()`

Everything is an object

- In object-oriented programming, all data are objects:
  - Variables, procedures, even libraries
- We make things happen by passing messages between objects
  - myFile.read(16)
  - appleName.upper()
- The object itself defines what messages it accepts: these are called its methods
  - e.g., files have read(), write(), etc.
    strings have upper(), len(), etc.



Methods and attributes

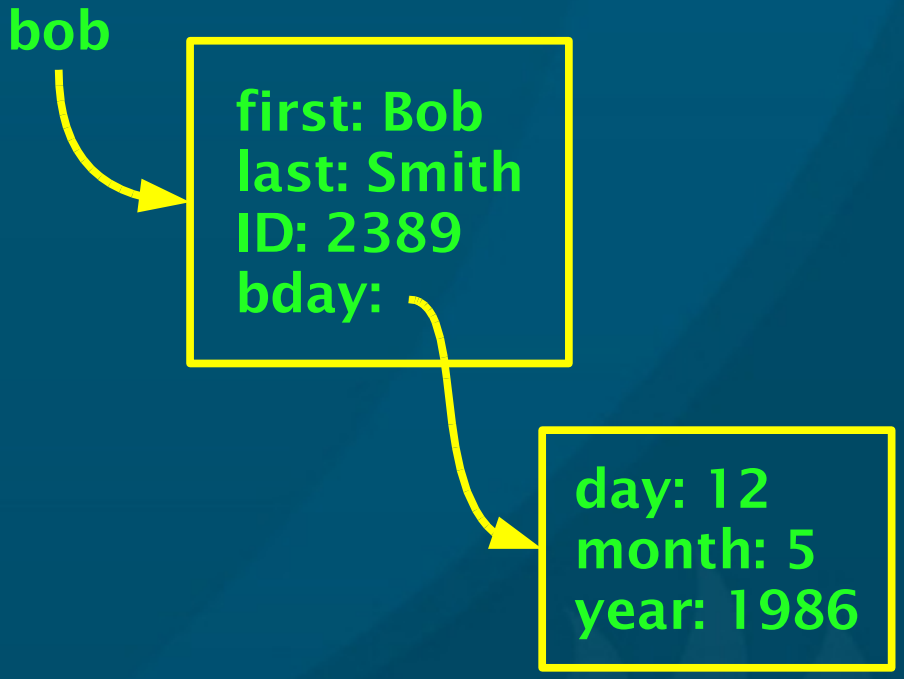
- Everything you can do with an object is encapsulated in its object definition
  - Methods make up the interface to the object
- Objects can also have attributes (variables)
- Our fractions.py ADT example:
  - Methods: get_n(), get_d(), add(), mult(), etc.
    - Everything you need to interact with a Fraction
  - Attributes: tuple (n,d)
    - Could also have two separate attributes: num, denom

Classes and instances

- We define (declare) object classes (types)
  - Attributes
  - Methods (interface)
    - Constructor and destructor
- Then we instantiate the class (declare variables)
- e.g., frac1 is a variable of type Fraction
  - frac1 is the instance,
  - Fraction is the class

More on instantiating classes

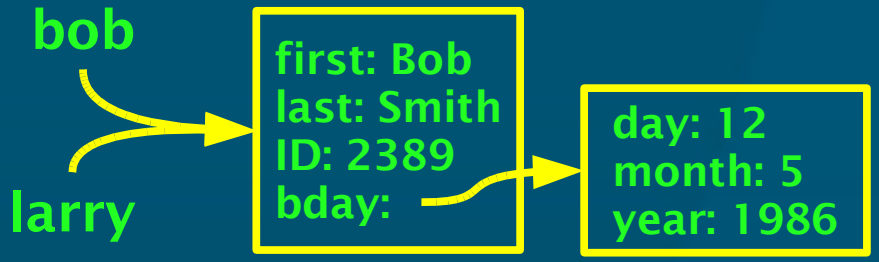
- class Date:
  - day = 0
  - month = 0
  - year = 0
- class StudentRecord:
  - firstName = ""
  - lastName = ""
  - ID = 0
  - birthdate = Date()



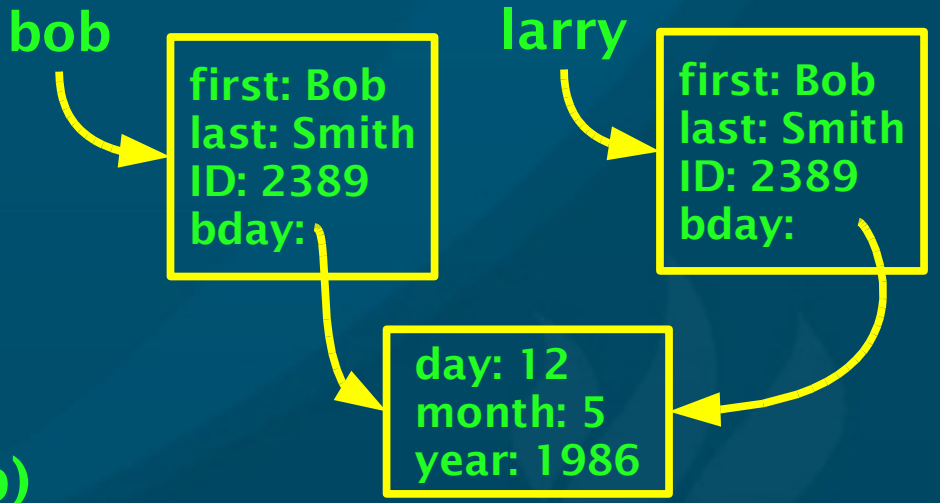
- Creating a new StudentRecord makes a new Date:
  - bob = StudentRecord()
  - bob.birthdate.year = 1986

More on copy vs. alias

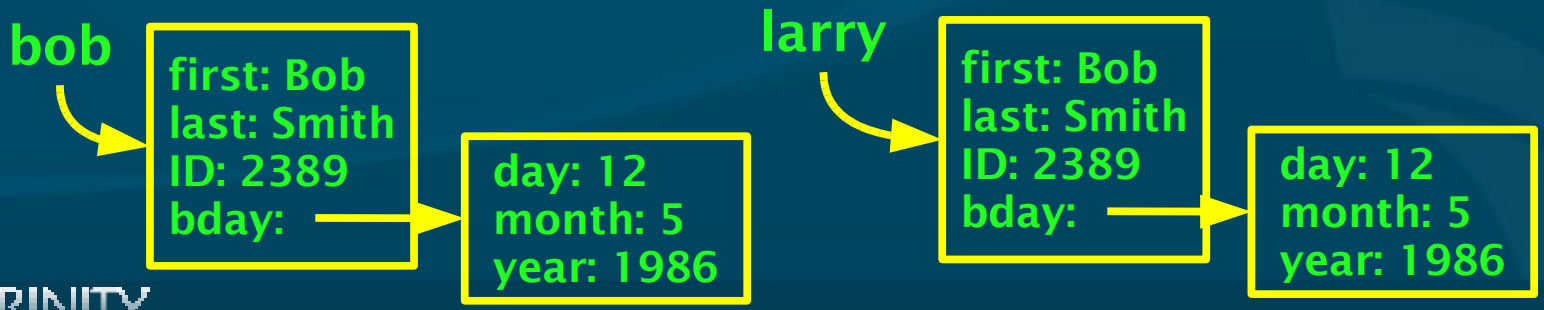
- Assignment: alias
  - larry = bob



- copy.copy(): shallow copy
  - larry = copy.copy(bob)



- copy.deepcopy(): deep copy
  - larry = copy.deepcopy(bob)



Review from today (Py ch12)

- Object-oriented programming paradigm
- Objects, methods, attributes
- Classes, instances
- Alias vs. shallow copy vs. deep copy

TODO

- No lab due this week!
- Lab07 due next week: Ch9 choose one:
  - #37+38: people db, matching
  - #40+41: online chequebook
  - #46: church directory
- HW08 due Wed:
  - Py ch12 # 3, 4
- Quiz08 (ch9) on Friday
- Paper topic by Mon 13Nov