Ling 555 — Programming for Linguists

Python — Object-oriented programming

Robert Albert Felty

Speech Research Laboratory
Indiana University

Oct. 27, 2008

Prereqs

While we are waiting

Please download the following files from:
http://robfelty.com/teaching/L555Fall2008/resources/py/
bird.py, penguin.py, happyfeet.py, lexicon.py

```
for file in
{bird.py,penguin.py,happyfeet.py,lexicon.py};
do curl -o $file
http://robfelty.com/teaching/L555Fall2008/
resources/py/$file; done
```

For Wednesday:

Read Chapter 8 on More Handling Exceptions

Outline

1. Homework questions
2. Programming paradigms
   - abstraction
3. Objects and Classes
   - intro
   - methods and attributes
   - toy examples
   - linguistic examples
   - tip
4. Theory
   - encapsulation
   - inheritance
   - polymorphism
5. Resources

Homework questions?

Programming paradigms

Three main types of programming paradigms (styles, genres, etc.)

imperative	Programs simply proceed one line at a time
functional	Programs consist mostly of functions
object-oriented	Programs are designed to mimic real-world objects

Levels of abstraction

You might wonder why we would use object-oriented programming, when it seems very abstract and unnecessary. Why not just use imperative or functional programming? Here are some reasons why:

- **Code re-use** We will use lexicons frequently. By creating a lexicon class, we can use this class for many different projects.
- **Structure** Object-oriented programming gives us more structure to our program, and in many cases makes the program more analogous to how we do things in the real world.
- **integration and collaboration** When we write programs in an object-oriented manner, it is very easy for others to write interfaces to our code, to write new methods for the same classes, or to write sub-classes which inherit our class

Objects and Classes

Definition

Class is to type as Object is to: _____

Objects and Classes

Definition

Class is to type as Object is to: _____

Example

- All cars have wheels and are self propelled.
  (class - describes car in general)
- Today I drove my car to work.
  (object - particular instance)

Methods and attributes

Definition

method (1) Something you can do with or to an object. (2) Function which is bound to a particular class. (3) verb

attribute (1) Property of a class. (2) noun/adjective

Example

- An attribute of a car is its color, or its engine type

```
mycar = Car() # create a new car object
mycar.color = 'silver'
mycar.engine = '4-cylinder'
```

- A method of a car is to drive, or to open a door

```
mycar.drive(to='Florida')
mycar.open(door='front-driver-side')
```

A bird class

Let's take a look at bird.py for a small example.

attributes

- song
- location

methods

- init()
- fly()
- sing()
- setSong()

A lexicon class

Let's take a look at lexicon.py to see how we can group together all the functions we created for working with dictionaries

attributes

- lexicon (stores the data)
- headers (stores the headers)

methods

- readFile()
- lookup()
- sorted()
- printWord()
- printDict()

Re-using your own code

Functions can call other functions

```
def readFile():
    fields=processLine(line)
```

Methods can call other methods

```
def printDict(self, file=sys.stdout):
    """ print out the entire
       dictionary to a file"""
    self.printHeaders()
    for word in self.lexicon.keys():
        self.printWord(word)
```

Encapsulation

Definition

- Attributes and methods of a particular object should not affect other objects. This is akin to the no-no of using global variables.
- All the inner workings of a class are said to be encapsulated.
- The rest of the world need only know how to use the various methods of the class, but not how they are implemented.
- This is similar to an API (Application Programming Interface).

Inheritance

Definition

- In the real world, there are classes and sub-classes (and sub-sub-classes etc.).
- For example, humans belong to the class of primates, which belongs to the class of mammals, which belongs to the class of vertebrates, which belong to the class of animals.
- Humans inherit attributes from these superclasses, such as the fact that humans have spines (inherited from vertebrates).

Polymorphism

Definition

- Flexibly defined classes can work with many different types of variables.
- This is refered to as polymorphism.
- For example, the + operator can add 2 integers, but can also concatenate strings.
- Polymorphism can be very powerful and handy, but it can also be tricky to implement.
- We won't worry about it too much.

Resources for you:

Some of the examples we have covered in class today can be found on the website at:
http://robfelty.com/teaching/L555Fall2008/resources/py
The files from today are:

1. bird.py
2. penguin.py
3. happyfeet.py
4. lexicon.py