Class 31: Python Dictionaries, Object-Oriented Programming

Upcoming Schedule

- **Reminder:** Peter has office hours right after class today!
- **Monday, 7 November:** Problem Set 6

Python Dictionaries

Dictionary abstraction provides a lookup table. Each entry in a dictionary is a <*key*, *value*> pair. The *key* must be an **immutable** object (so it cannot be a (mutable) list). The *value* can be anything.

dictionary[*key*]
evaluates to the *value* associated with *key*. Running time is approximately constant!

dictionary[*key*] = *value*
If *key* is already in the dictionary, updates the value associated with *key* to *value.*
Otherwise, creates a new <*key*, *value*> entry in the dictionary.

Using Dictionaries

```
def histogram(text):
  d = {}
  words = text.split()
  for w in words:
    if w in d:
      d[w] = d[w] + 1
    else:
      d[w] = 1
  return d
```

What is the running time of histogram?

Lambda in Python

Expression ::= **lambda** *Parameters* : *Expression*

```
(lambda a, b: a > b) (3, 4)
```

sorted(*collection*, *cmp*)
Returns a new sorted list of the elements in *collection* ordered by *cmp*.
cmp specifies a comparison function of two arguments which should return a negative, zero or positive number depending on whether the first argument is considered smaller than, equal to, or larger than the second argument.

```
def show_histogram(d):
   keys = d.keys()
   okeys = sorted(keys, lambda k1, k2: d[k2] - d[k1])
   for k in okeys:
      print str(k) + ": " + str(d[k])
```

Object-Oriented Programming

What is an *object*?

What is *object-oriented programming*?

How do Smalltalk-ers add 2 + 3?

Why is it useful to package state and procedures together?

Who was the first object-oriented programmer?

If you have an idea, and its not a good idea, take a nap instead of implementing it.
Alan Kay (note: this only works if you start your assignments early!)

The people who are the worst at programming are the people who refuse to accept the fact that their brains aren't equal to the task. Their egos keep them from being great programmers. The more you learn to compensate for your small brain, the better a programmer you'll be. The more humble you are, the faster you'll improve.
Edsger Dijkstra, 1972 Turing Award Speech

I don't know how many of you have ever met Dijkstra,
but you probably know that arrogance in computer science is measured in nano-Dijkstras.
Alan Kay