CS1110

Lecture 22: Prelim 2 Review Session

Announcements

Processed prelim regrade requests: on the front table.

Reminders:
Exam: 7:30–9:00PM, Tuesday Apr 16th Kennedy 116 (Call Auditorium, same as before). Arriving early recommended.

Next Tuesday: lectures replaced by professor office hours in Thurston 102; and labs are staff office hours. Next Wed-Fri: all labs, staff and consulting hours canceled.

We will start (and, we hope, finish) grading Thursday evening.

Additional sources of practice problems

- Fall 2012 Prelim 2, questions 4, 5.
- Fall 2012 Final questions, questions 4, 6, 7 given the invariant for insertion sort and the helper function push_down.
- The worked exercises on loop invariants
- Loop problems at http://codingbat.com/python (interactive programming exercises). You can try solving the problems via both loops and recursion, often.

Material emphasized

Recursion (A4, Lab 6)
Defining and using classes (A4, A6, Lab 8)
For- and while-loops (A4, A6, Lab10)
Code development from invariants (A6, Lab10)

[We will assume knowledge of the material covered previously, in the sense that you should know how to, say, create lists, draw folders and frames, manipulate strings, make appropriate test cases, and so on. This shouldn't require additional studying.]

Provide a recursive implementation

```python
def merge(s1, s2):
    """Returns: characters of s1 and s2, in alphabetical order.
Examples: merge('ab', '') = 'ab'
merge('abbce', 'cdg') = 'abbccdeg'
Precondition: s1 a string with characters in alphabetical order
s2 a string with characters in alphabetical order"
```

Provide a for-loop implementation

```python
def skip(s):
    """Returns: copy of string s, odd letters dropped.
Example: 'abcd' -> 'bd'. ' ' -> ' '  'abc' -> 'b' ""
```
Provide a while-loop implementation

```python
def skip(s):
    """Returns: copy of string s, odd letters dropped.
    Example: 'abcdef' -> 'bdf'.  """  
```

Defining a class

```python
class Paper(object):
    """An instance is a scientific paper.
    Class variables:
    number [int]: number of papers that have been created.  >= 0
    Instance variables:
    title [string]: title of this paper.  At least one char long.
    cites [list of Papers]: papers that this book cites
    cited_by [list of Papers]: papers that this paper is cited by
    """
    number = 0  # Initial value is 0
```

Implement according to invariant

```text
- Given a sequence b[h..k] with some value x in b[h]:
  
  pre: b
  x
  k

  post: b
  <= x
  x
  => x

  law: b
  <= x
  x
  => x
```

```python
def evaluate(p, x):
    """Returns: The evaluated polynomial p(x).
    We represent polynomials as a list of floats:
    [1.5, -2.2, 3.1, 0, -1.0] is 1.5 - 2.2x + 3.1x^2 + 0x^3 - x^4
    We evaluate by substituting in for the value x. For example
    evaluate([1.5, -2.2, 3.1, 0, -1.0], 2) = 1.5 - 2.2(2) + 3.1(4) - 1(16) = -6.5
    evaluate([3], 4) = 2
    Precondition: p is a list (len > 0) of floats, x is a float"
```