CS 1110

Prelim 1 Review
Fall 2012
Exam Info

- Prelim 1: 7:30–9:00PM, Thursday, October 4th
  - Last name **A – P** in Kennedy 1116
  - Last name **R – T** in Warren 131
  - Last name **U – Z** in Warren 231
- To help you study:
  - Study guides, review slides are online
  - Solutions to Assignment 2 are in CMS
- Arrive early! Helps reducing stress
- Grades released the same evening (if possible)
What is on the Exam?

• Five Questions (+2pts for name, netid):
  ▪ String slicing functions (A1)
  ▪ Call frames and the call stack (A2)
  ▪ Functions on mutable objects (A3)
  ▪ Testing and debugging (A1, Lab 3)
  ▪ Short Answer (Terminology)

• Roughly equal weight each
What is on the Exam?

• String slicing functions (A1)
  § Will be given a function specification
  § Implement it using string methods, slicing
• Call frames and the call stack (A2)
• Functions on mutable objects (A3)
• Testing and debugging (A1, Lab 3)
• Short Answer (Terminology)
def make_netid(name, n):
    """Returns a netid for name with suffix n
    Netid is either two letters and a number (if the student has no
    middle name) or three letters and a number (if the student has
    a middle name). Letters in netid are lowercase.
    Example: make_netid('Walker McMillan White', 2) is 'wmw2'
    Example: make_netid('Walker White', 4) is 'wmw4'
    Precondition: name is a string either with format '"<first-name>"<last-name>"' or '"<first-name> <middle-name> <last-name>"';
    names are separated by spaces. n > 0 is an int."""
### Useful String Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>s.find(s1)</td>
<td>Returns first position of s1 in s; -1 if not there.</td>
</tr>
<tr>
<td>s.rfind(s1)</td>
<td>Returns first position of s1 in s; -1 if not there.</td>
</tr>
<tr>
<td>s.lower()</td>
<td>Returns copy of s with all letters lower case</td>
</tr>
<tr>
<td>s.upper()</td>
<td>Returns copy of s with all letters upper case</td>
</tr>
</tbody>
</table>

- We will give you any methods you need
- But you must know how to slice strings!
What is on the Exam?

• String slicing functions (A1)
• Call frames and the call stack (A2)
  - Very similar to A2 (see solution in CMS)
  - May have to draw a full call stack
  - See lectures 5 and 9 (slide typos corrected)
• Functions on mutable objects (A3)
• Testing and debugging (A1, Lab 3)
• Short Answer (Terminology)
Call Stack Example

Given functions to right
- Function `fname()` is not important for problem
- Use the numbers given

Execute the call: `lname_first('John Doe')`

Draw *entire* call stack when helper function `lname` completes line 1
- Draw nothing else

```python
def lname_first(s):
    """Precondition: s in the form <first-name> <last-name>""
    first = fname(s)
    last = lname(s)
    return last + ',' + first

def lname(s):
    """Prec: see last_name_first""
    end = s.find(' ')  
    return s[end+1:]
```

9/30/12 Prelim 1 Review
Example with a Mutable Object

```python
def shift(p):
    """Shift coords left
    Precondition: p a point""
    temp = p.x
    p.x = p.y
    p.y = p.z
    p.z = temp

>>> p = Point(1.0,2.0,3.0)
>>> shift(p)
```

- May get a function on a mutable object
  ```python
  >>> p = Point(1.0,2.0,3.0)
  >>> shift(p)
  ```
- You are not expected to come up w/ the “folder”
  - Will provide it for you
  - You just track changes
Example with aMutable Object

def shift(p):
    """Shift coords left
    Precondition: p a point""
    temp = p.x
    p.x = p.y
    p.y = p.z
    p.z = temp

>>> p = Point(1.0,2.0,3.0)
>>> shift(p)
What is on the Exam?

• String slicing functions (A1)
• Call frames and the call stack (A2)
• Functions on mutable objects (A3)
  ▪ Given an object type (e.g. class)
  ▪ Attributes will have invariants
  ▪ Write a function respecting invariants
• Testing and debugging (A1, Lab 3)
• Short Answer (Terminology)
Example from Assignment 3

• Type: RGB
  ▪ Constructor function: RGB(r,g,b)
  ▪ Remember constructor is just a function that gives us back a mutable object of that type
  ▪ Attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Invariant</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>int, within range 0..255</td>
</tr>
<tr>
<td>green</td>
<td>int, within range 0..255</td>
</tr>
<tr>
<td>blue</td>
<td>int, within range 0..255</td>
</tr>
</tbody>
</table>
def lighten(rgb):
    """Lighten each attribute by 10%
    Attributes get lighter when they increase.
    Precondition: rgb an RGB object"
    pass # implement me
Another Example

- **Type:** Length
  - **Constructor function:** Length(ft,in)
  - Remember constructor is just a function that gives us back a mutable object of that type
  - **Attributes:**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Invariant</th>
</tr>
</thead>
<tbody>
<tr>
<td>feet</td>
<td>int, non-negative, = 12 in</td>
</tr>
<tr>
<td>inches</td>
<td>int, within range 0..11</td>
</tr>
</tbody>
</table>
Function that Does Not Modify Object

```python
def difference(len1, len2):
    """Returns: Difference between len1 and len2
    Result is returned in inches
    Precondition: len1 and len2 are length objects
    len1 is longer than len2""
    pass  # implement me
```

What is on the Exam?

• String slicing functions (A1)
• Call frames and the call stack (A2)
• Functions on mutable objects (A3)
• Testing and debugging (A1, Lab 3)
  ▪ Coming up with test cases
  ▪ Tracing program flow
  ▪ Understanding asserts, try-except
• Short Answer (Terminology)
def pigify(w):

    """Returns: copy of w converted to Pig Latin
'y' is a vowel if it is not the first letter
If word begins with a vowel, append 'hay'
If word starts with 'q', assume followed by 'u';
move 'qu' to the end, and append 'ay'
If word begins with a consonant, move all
consonants up to first vowel to end and add 'ay'
Precondition: w contains only (lowercase) letters"
"""
def first(x):
    print 'Starting first.'
    try:
        second(x)
    except:
        print 'Caught at first'
    print 'Ending first'

def second(x):
    print 'Starting second.'
    try:
        third(x)
    except:
        print 'Caught at second'
    print 'Ending second'

def third(x):
    print 'Starting third.'
    assert x < 1
    print 'Ending third.'

What is the output of first(2)?
def first(x):
    print 'Starting first.'
    try:
        second(x)
    except:
        print 'Caught at first'
    print 'Ending first'

def second(x):
    print 'Starting second.'
    try:
        third(x)
    except:
        print 'Caught at second'
    print 'Ending second'

def third(x):
    print 'Starting third.'
    assert x < 1
    print 'Ending third.'

What is the output of first(0)?
What is on the Exam?

• String slicing functions (A1)
• Call frames and the call stack (A2)
• Functions on mutable objects (A3)
• Testing and debugging (A1, Lab 3)
• Short Answer (Terminology)
  ▪ See the study guide
  ▪ Look at the lecture slides
  ▪ Read relevant book chapters

In that order