CS 1110
Prelim 1 Practice/Review Session

Exam Topics

- String slicing functions
- Call frames and the call stack
- Functions on mutable objects
- Testing and debugging
- Conditionals
- Lists and simple iteration

Lists, Iteration, Strings

```python
def count_non_space_chars(myList):
    # Returns: number of non-space characters in the strings in myList.
    # Example: count_non_space_chars(['U', 'r', '', 'gr8']) returns 5
    # Precondition: myList is a list of strings. Each string in myList can contain only spaces, letters, digits.

    count = 0
    for s in myList:
        numSp = s.count(' ')  # count spaces
        numNonSp = len(s) - numSp  # total length minus spaces
        count += numNonSp
    return count
```

Lists, Iteration, Types

```python
def inflate(myList, p_percent):
    # Inflates each number in myList by p_percent while maintaining the type (int or float).
    # For any int in myList, round down the inflation. Precondition: myList is a list of positive numbers (int and/or float).
    # Precondition: p_percent is a positive number (int or float).

    # An example:
    >>> aList= [100, 100.0, 1, 1.0]
    >>> p= 1.6
    >>> inflate(aList,p)
    >>> aList
    [101, 101.6, 1, 1.016]
```

Announcements

- A3 due Sun Mar 28
- Prelim 1 Tues Mar 30 at 6:30pm in-person (university-scheduled)
- Check CMS for your exam info if you requested alternate time/format
- In-person: Bring pens/pencils/erasers (bring several). Bring a watch or even an actual clock if you have one. No smart watches/phones! You may not be able to see the wall clock in Barton from your seat. Bring Cornell ID.
- Online: Your proctor will contact you about a mock exam. You must do the mock exam to be allowed to write the actual exam.
- Read Prelim 1 Study Guide. Note spring different from fall.
- Tues Mar 30 lecture and lab time ➔ office hours
- Wedn Mar 31 no labs (so no new lab exercises next week)
def inflate(myList, p_percent):
    """Inflate each number in myList by p_percent while maintaining the type (int or float). For any int in myList, round down the inflation. Precondition: myList is a list of positive numbers (int and/or float). Precondition: p_percent is a positive number (int or float)."""
    p_frac = p_percent / 100
    for k in range(len(myList)):
        delta = myList[k] * p_frac
        if type(myList[k]) == int:
            delta = int(delta)
        myList[k] += delta

def before_space(s):
    """Returns: the substring before the first space character in string s. Precondition: string s contains at least one space."""
    return s

Examples:
- "abc" → "" → single space char at the start
- "abc" → "abc" → single space char at the end
- "a bc" → "a" → single space char in the "middle" (not start or end)
- " abc" → " " → many space chars at the start
- "abc " → "abc" → many space chars at the end
- "ab c" → "ab" → many space chars in the middle
- "a b c" → "a" → many non-adjacent space chars

Remember:
- Give yourself an example if question doesn't provide one
- Using for-loop on list: do you need to modify list? If so you need the indices—use range
- List syntax
- How to work with types (ops, checking, casting)
- In general, read specs again after finishing code. Did you really solve problem asked?

Common Cases: typical usage
Edge Cases: live at the boundaries
- Target location in list: first, middle, last elements
- Input size: 0, 1, 2, many (length of lists, strings, etc.)
- Input Orders: e.g., max(big, small), max(small, big)…
- Element values: negative/positive, zero, odd/even
- Element types: int, float, str, etc.
- Expected results: negative, 0, 1, 2, many

Not all categories/cases apply to all functions. Use your judgement!
Functions on Objects

• Class: Rect
  • Constructor function: Rect(x, y, width, height)
  • Remember constructor is just a function that gives us an object of that type and returns its identifier

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>float, x coord of lower left corner</td>
</tr>
<tr>
<td>y</td>
<td>float, y coord of lower left corner</td>
</tr>
<tr>
<td>width</td>
<td>float, &gt; 0, width of rectangle</td>
</tr>
<tr>
<td>height</td>
<td>float, &gt; 0, height of rectangle</td>
</tr>
</tbody>
</table>

Remember:
• Draw a diagram to help yourself think
• Label the diagram with example/known values.
• Then generalize labels using parameter and attribute names.
• Important problem solving step! First use example values to understand the problem and figure out relationships among knowns and unknowns.
• Dot-notation for accessing attributes of an object

def move(r, xc, yc):
    """Set the attributes of Rect `r` such that its center lies on the x- and y-coordinates `xc` and `yc`, respectively.
    Precondition: r is a Rect object.
    Precondition: xc, yc are each a float."

    r.x = xc - r.width/2
    r.y = yc - r.height/2

8
(20,50)
6
(20-8/2, 50-6/2)

8
(20,50)
6
(20-8/2, 50-6/2)

Good Luck!