# CSC108H Lecture 16

Dan Zingaro

October 17, 2012

# Tuples

- Like lists
  - Sequences, subscript, slices
- Not like lists
  - Only `count` and `index` methods
  - Created with parentheses (not brackets)
  - Immutable
- Provide data integrity (e.g. no possible changes through aliasing)

# Uppercase List and While

Use a while-loop to write both of these:
(1) Given a list of strings, write a function that returns a list that contains all of these strings in uppercase. Do **not** modify the original list.
(2) Given a list of strings, write a function that modifies the list so that all strings are in uppercase. Do **not** return the list.

# ConcepTest

```
def find(lst, value):
  '''(list, value) -> int
  Return the first occurrence of value in lst.
  If value is not found, return -1.

  >>> find([20, 40, 60], 40)
  1
  '''
  i = 0
  num = lst[i]
  while num != value:
    i = i + 1
    num = lst[i]
  if i < len(lst):
    return i
  return -1
```

In what situation does the above code fail?

- A. It never fails
- B. It fails when the list is empty
- C. It fails when the value is not found in the list
- D. Both B and C will fail

# Example 1: Find

```
def find(lst, value):
  '''(list, value) -> int
  Return the first occurrence of value in lst.
  If value is not found, return -1.

  >>> find([20, 40, 60], 40)
  1
  '''
```

Several ways to solve this:

1. Using a boolean operator in the while-loop condition
2. Using a `return` inside the loop

# ConcepTest

To represent groups of students, we can use nested list. For example, in the following list, students 1, 3, and 4 are together in a group, and student 2 is working alone:
`[[1, 3, 4], [2]]`

```
def is_ok(group_list, class_list):
  '''(list of list of int, list of int) -> bool
  Return True iff every student in class_list is in exactly
  one group according to group_list.
  '''
```

Which call would return `True`?

- A. `is_ok([[1, 2], [2, 3]], [1, 2, 3])`
- B. `is_ok([[1, 2], [4]], [1, 2, 3])`
- C. `is_ok([], [1, 2, 3])`
- D. `is_ok([[1, 2], [4]], [1, 2, 3, 4])`
- E. None will return `True`

# Example 2: Student Groups

```
def is_ok(group_list, class_list):
    '''(list of list of int, list of int) -> bool
    Return True iff every student in class_list is in exactly
    one group according to group_list.

    >>> is_ok([[1, 3, 4], [2]], [1, 2, 3, 4])
    True
    '''
```

# Example 3: Column Sums

```
def sums(lst):
    '''(list of list of int) -> list of int
    Return a new list that contains the sum of each column
    in lst. All sublists are of the same length.

    >>> sums([[5, 10, 15], [1, 2, 3]]
    [6, 12, 18]
    '''
```