Lab 8

- List Review
- Deal or No Deal

Lab 7 Feedback

- Missing function comments
  - Others need to know *how to use* the function
  - Good comments in lab description
- Extra step in game module's `flipCoin` function

```
HEADS=0
TAILS=1

def flipCoin():
    return random.randint(0,1)
```

Equivalent code (needs comments)

```
HEADS=0
TAILS=1

def flipCoin():
    flip = random.randint(0,1)
    if flip == HEADS:
        return HEADS
    else:
        return TAILS
```


Comment Example

```
# Encodes a single character.
# PRE: Input parameters are a single, lowercase
# character string (char) and an integer key
# (between -25 and 25, inclusive)
# POST: returns the encoded character
def translateLetter(char, key):
```

- Does not say *who* called function, where returned to
- Does not say variable name returned

Commenting Exercise

- Write a comment for this function:

```
def encode(toEncode, key):
```


Commenting Exercise

- Write a comment for this function:

```
# Encodes a lowercase string using a key,
# preserving spaces in the string.
# PRE: a lowercase string to be encoded
# (toEncode) and the integer key (between -25
# and 25, inclusive)
# POST: returns the encoded string
def encode(toEncode, key):
```


Commenting Notes

- Well-named parameters make documentation easier
- I'm not strict on the pre/post format.
- Just need to be clear on
  - what the function does (at high level)
  - types of parameters
  - type of the output

➔ The caller knows what to pass to the function and if they should assign the output to a variable

Review

- True or False: If we modify a *list* variable inside a function, it is modified outside of the function
- True or False: If we modify an *integer* variable inside a function, it is modified outside of the function

Passing Lists as Parameters

- Lists are **mutable**
- Can be modified inside of functions
  - Modifications seen outside of functions
- Example from yesterday:
  - `descendSort3Nums( list3 )`

Modifying Lists Passed as Parameters

```
def descendSort3Nums(list3):
    if list3[1] > list3[0]:
        # swap 'em
        tmp = list3[0]
        list3[0] = list3[1]
        list3[1] = tmp

    if list3[2] > list3[1]:
        tmp = list3[1]
        list3[1] = list3[2]
        list3[2] = tmp

    if list3[1] > list3[0]:
        tmp = list3[0]
        list3[0] = list3[1]
        list3[1] = tmp
```

```
def main():
    myList = [1,2,3]
    descendSort3Nums(list)
    print myList
```

`myList` gets modified by function

Note: Function does not *return* anything. Simply modifies the `list3` parameter.

Rules for Collaboration

- Debugging help
  - 5 minute rule: a friend can only look at your code *to help with debugging* for 5 minutes
  - Owner of code owns keyboard/mouse
- Problem solving discussion
  - No written solutions leave the room
- Acknowledge aid
  - Including from Camille and Will, outside of lab
- Do not give out your password

Deal or No Deal Functionality

- Read in values contained in cases from a file
  - What data type should these numbers be?
- Have user select from remaining cases
  - Make sure choice is valid
- Display remaining cases
  - Print four to a row
- Display remaining amounts on board
  - Left column is smaller amounts

Exam 2

- Mean: 82%, Median: 88%
- Change in performance from first exam
  - Better: 4
  - Same (+/- 2): 5
  - Worse: 12
- Last 4 pages of problems: best indicators of understanding
- Effect on final course grade
  - Combined exams worth 30%
  - Weighted toward higher grade (e.g., 16/14)

Data Types of Loop Variables

For Loop Header	Data Type of x
`for x in <file>:`	`string` (line in a file)
`for x in xrange(…):`	`int`
`for x in <string>:`	`string` (character)
`for x in <list>:`	Depends on elements in list

Benefit of OO Programming

- *Not* a benefit of functions
  - Bonus 1 point for "packages data and methods/ operations together"

Averaging Grades

- Similar to averaging temperatures problem in class and Olympics problem in lab

```
majorsFile = file("majors.dat", "r")
total = 0
for numMajors in majorsFile:
    total += numMajors
majorsFile.close()
print "The total number of majors is", \
    numMajors
```

Can be given any name

numMajors needs to be converted to a *number* to add to **total**

Converting `while` loop

```
MIN = 0
x = 6
a = 0
while x >= MIN:
    print "x is", x
    a += x
    x -= 2
print "a is", a
```

Bonus point for recognizing how x can be decremented by `for` loop

```
MIN = 0
a = 0
for x in xrange(6, MIN-1, -2):
    print "x is", x
    a += x
print "a is", a
```


Tracing Through Functions

- Only *one thing* gets returned from function
- Trace through call to

  `helper1("springtime", "i")`

```
def helper1(word, letter):
    for i in xrange(len(word)):
        if word[i] == letter:
            return i
    return -1
```

What does `i` represent?
When is `i` returned? When is -1 returned?

Tracing Through Functions

- Only *one thing* gets returned from function

```
def helper1(word, letter):
    for i in xrange(len(word)):
        if word[i] == letter:
            return i
    return -1
```

Name: **`firstPos`**
Comment: returns the *position* of the *first* occurrence of the given letter in the word; returns -1 if the letter does not occur in the word

Lab 8 Overview

- List problems
- *Deal or No Deal*