Today's "Menu"
(week 7)

- Announcements
- IDLE
- Brief review of while-/for-loop
- **Nested loops** (very important and common)
- Questions?

Announcements

- Exam II coming up soon …

IDLE

- Explore your environment
- Look at the options

- Aside: Talk to yourself if you get stuck!

Loops

- 2 types: for and while
- Use for if we know ahead of times how many iterations
- Use while if we don't know ahead of times how many iterations
- Both loops are equivalent – for does some 'housekeeping' for us.
- Learn to use the range() function

Review - while

```
regular program code

"set up" condition to enter loop (or not) # counter/Boolean
while <BE>:
      body of loop
      more body of loop
      do something to exit loop  # change counter/Boolean

regular code …
```

Suggestion: Get basic loop control setup first, _then_ worry about other parts "inside" loop.

Review - for

- Usually used with range() function
- Use when we know ahead of time how many times we want to execute something
- Does a lot of 'housekeeping' for us

```
for i in range(5):
    print '%d * %d = %d' % (i, i, i * i)
```

More examples …

```
def printNumbers(num):
  for j in range(num):
        print j,
```

Output??

More examples …

```
def printNumbers(num):
    # method to print numbers
    for j in range(num):
        print j,

def main():
    # main loop
    for i in range(5):
        printNumbers(12)
        print '*'
main()
```

Output??

Nested loops (1)

- Loops with a loop

```
for i in range(5): #outer loop starts
    for j in range(12): # inner loop starts
        print j, # inner loop executes + ends
    print '*' # outer loops executes + ends
```

- Inner loop gets executed completely each time through the outer loop.
- Output?

Nested loops (2)

- Loops with a loop

```
for i in range(5):
    for j in range(12):
        print j,
    print '*'
```

```
0 1 2 3 4 5 6 7 8 9 10 11 *
0 1 2 3 4 5 6 7 8 9 10 11 *
0 1 2 3 4 5 6 7 8 9 10 11 *
0 1 2 3 4 5 6 7 8 9 10 11 *
0 1 2 3 4 5 6 7 8 9 10 11 *
```

Be sure you understand this …

```python
def printNumbers(num):
    # method to print numbers
    for j in range(num):
        print j,

def main():
    # main loop
    for i in range(5):
        printNumbers(12)
        print '*'
main()
```

Output??

and this …

- Loops with a loop

```python
for i in range(5): #outer loop starts
    for j in range(12): # inner loop starts
        print j, # inner loop executes + ends
    print '*' # outer loops executes + ends
```

- Inner loop gets executed completely each time through the outer loop.
- Output?

For loops

http://cs.franklin.edu/~esmail/COMP_480/SampleProgs/for_loops.py

Summary/Recap

• Questions?