

Objectives

- Advanced problem solving with for loops

Jan 23, 2008

Sprenkle - CS111

1

Lab Review

- UNIX Network utilities
- With a little more training (e.g., automation, interfaces), you'd be able to write WikiScanner!

Jan 23, 2008

Sprenkle - CS111

2

for loop review

```
for x in xrange(5):  
    # like assigning x values(0,1,2,3,4)  
  
    # rest of loop body ...
```

- Note: when have xrange(5)
 - x gets values (0, 1, 2, 3, 4)
 - Which means that loop executes 5 times

Jan 23, 2008

Sprenkle - CS111

3

Practicing for Loops

- Print the following:

➤ A) 1 2 3 4 5

➤ B) 2 5 8 11

➤ C) ****

Jan 23, 2008

Sprenkle - CS111

4

Practicing for Loops

- Print the following:

```
1 2 3 4 5  
1 2 3 4 5  
1 2 3 4 5
```

Jan 23, 2008

Sprenkle - CS111

5

Nested for Loops

- Use when need to repeat a loop
 - Good programming practice: use different variables for inner and outer loop variable

```
for x in xrange(N):           Outer loop  
    statementa
```

```
        for y in xrange(M):   Inner loop  
            statementb
```

- Analysis: how many times are **statementa** and **statementb** repeated?

Jan 23, 2008

Sprenkle - CS111

6

Fence Post Problem

- Given some posts and some beams to connect the posts, build a fence that is X fenceposts long

Posts: |
Beams: -

Jan 23, 2008

Sprenkle - CS111

fence_post.py

7

Practice: Draw a Tic-Tac-Toe Board

```
| | |  
- - -  
| | |  
- - -  
| | |
```

Jan 23, 2008

Sprenkle - CS111

tictactoe.py

8

Practice: Assign Students to Groups

- Using a **for** loop and the modulo (%) operator, assign students to groups
 - How would you “model” students (given the above problem specification)?

Jan 23, 2008

Sprenkle - CS111

9

Practice: Assign Students to Groups

- Using a **for** loop and the modulo (%) operator, assign students to groups
 - How would you “model” students (given the above problem specification)?

```
numStudents = 12  
numGroups = 4  
for student in xrange(numStudents):  
    print student, "is in group",  
    print (student%numGroups)+1
```

Jan 23, 2008

Sprenkle - CS111

10

random module

- Python provides the **random** module to generate pseudo-random numbers
- Why “pseudo-random”?
 - Actually generates a list of numbers and grabs the next one off the list
 - A “seed” is used to initialize the random number generator
 - By default, the current time is used as the seed

Jan 23, 2008

Sprenkle - CS111

11

Some random Functions

- random()**
 - Returns the next random floating point number in the range [0.0, 1.0)
- randint(a, b)**
 - Return a random integer N such that $a \leq N \leq b$

```
import random  
  
#random.seed(1)  
  
for x in xrange(10):  
    print random.random()
```

Jan 23, 2008

Sprenkle - CS111

random_test.py

12

VA Lottery: Pick 4

- To play: you pick 4 numbers between 0 and 9
- To win: select the numbers that are selected by the magic ping-pong ball machine
- Your job: Simulate the magic ping-pong ball machines
 - Display the number on one line

Jan 23, 2008

Sprenkle - CS111

13

VA Lottery: Pick 4

- To play: you pick 4 numbers between 0 and 9
- To win: select the numbers that are selected by the magic ping-pong ball machine
- Your job: Simulate the magic ping-pong ball machines
 - Revision: display number as ~~##-##~~##

Jan 23, 2008

Sprenkle - CS111

[pick4.py](#)

14