

Objectives

- Lab Review
- Security issues
- Broader Issue: Facebook's Newsfeed and Facebook's Success

Nov 16, 2007

Sprenkle - CS111

1

Lab Review

- **Main goal:** Use **python** to enable analysis for a scaled-down scientific application
- Creating, using classes
- (Large) File processing, writing
- Dictionaries with objects
- Harder to test
- A free graphing application (gnuplot)
- Task automation
 - May want to analyze requests every night
- Common optimization technique: "caching"

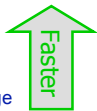
Nov 16, 2007

Sprenkle - CS111

2

Lab Review: Caching

- Cache: commonly used in computer science
- Think of flying in an airplane:
 - Seat pocket
 - Under seat
 - Overhead bin
 - Checked luggage
- "Cached" hostnames for IP addresses in dictionary
 - Look up in dictionary (cheap) so that don't have to look up again using UNIX command (expensive)



Nov 16, 2007

Sprenkle - CS111

3

Lab Review: Caching

- DNS cached too
 - Keep frequently accessed IP/hostnames so don't have to look up again.

Nov 16, 2007

Sprenkle - CS111

4

Security Considerations

- We've been trying to make our programs very flexible, general
 - E.g., User specifies input (file names); take command-line arguments
- Must validate user input
 - We have done some of this but we can do more
 - Need to improve restriction of user input
 - Microsoft security/reliability architect recommends that you restrict the sources of input (gets rid of trust issues)

Nov 16, 2007

Sprenkle - CS111

5

Example Program

```
def main():
    #Program mission statement
    print "This program determines your birth year"
    print "given your age and the current year\n"

    age=input("Enter your age: ")
    currentyear=input("Enter the current year: ")

    #Subtract age from current year
    birthyear=currentyear - age
    #Display output to the user
    print "You were either born in", birthyear, "or", birthyear-1
```

What can we do to restrict the user's input?

Nov 16, 2007

Sprenkle - CS111

6

Validating User Input (Better)

```
def main():
    #Program mission statement
    print "This program determines your birth year"
    print "given your age and the current year \n"

    age=input("Enter your age: ")
    currentyear=input("Enter the current year: ")

    if age < 0 :
        print "Come on: you need to have a positive age."
    else:
        #Subtract age from current year
        birthyear=currentyear - age
        #Display output to the user
        print "You were either born in", birthyear, "or", birthyear-1
```

Nov 16, 2007

Sprenkle - CS111

birthyear.py

7

Validating User Input

- What happened when the user entered something like "B6"?

Nov 16, 2007

Sprenkle - CS111

8

Validating User Input

- What happened when the user entered something like "B6"?
 - Threw an Exception and the program exited

```
Enter your age: B6
Traceback (most recent call last):
  File "currentAge.py", line 22, in <module>
    main()
  File "currentAge.py", line 9, in main
    age=input("Enter your age: ")
  File "<string>", line 1, in <module>
NameError: name 'B6' is not defined
```

Nov 16, 2007

Sprenkle - CS111

9

Handling Exceptions

- Using try/except statements
- Syntax:

```
try:
    <body>
except [<errorType>]:
    <handler>
```

- Example:

```
try:
    age = input("Enter your age: ")
    currentyear = input("Enter the current year: ")
except:
    print "ERROR: Your input was not in the correct form."
    print "Enter integers for your age and the current year"
    return
```

Nov 16, 2007

Sprenkle - CS111

birthyear2.py

10

Handling Exceptions

- Could put try/catch statements in a loop to make sure user enters valid input
 - Example: birthyear3.py
- Other types of exceptions
 - File doesn't exist
 - Could prompt for new file
 - Don't have permission to read/write file
 - Could prompt for new file

Nov 16, 2007

Sprenkle - CS111

11

Another Example of Security Bug

- Example from a real web site
 - Simulated in web pages in class
- More fundamental than a security bug
 - How to address this security issue?
 - Check that number of books is positive!

Nov 16, 2007

Sprenkle - CS111

12

Example of Security Bug in Python

- Demonstrate with security_bug.py

Nov 16, 2007

Sprenkle - CS111

13

Why the Security Bug?

- `input` is actually `eval(raw_input(...))`
- How to fix?
 - In the future, only `raw_input` will be allowed
 - Fixed code: fixed_input.py

Nov 16, 2007

Sprenkle - CS111

14

File Input from User

- Alternative 1:
 - Prompt user: What file do you want the program to process?
 - Prompt user: What file do you want the program to write/output?

Nov 16, 2007

Sprenkle - CS111

15

File Input from User

- Alternative 1:
 - Prompt user: What file do you want the program to process?
 - Prompt user: What file do you want the program to write/output?
- **Issues:**
 - What if bad input file? (not just bad name)
 - What if output file writes over existing file?
 - May or may not be malicious; could just be dippy

Nov 16, 2007

Sprenkle - CS111

16

File Input from User

- **Issues:**
 - What if bad input file? (not just bad name)
 - What if output file writes over existing file?
 - May or may not be malicious; could just be dippy
- How to address issues?
 - Check if (input/output) file exists first
 - Use the `os` module
 - Permission problems should be handled by OS

Nov 16, 2007

Sprenkle - CS111

17

File Input from User (Safer)

- Alternative 2:
 - Prompt user: Which of x files do you want the program to process?
 - The x files are known to be okay
 - Automatically create (valid) file name
 - Again, check that file does not already exist

Nov 16, 2007

Sprenkle - CS111 file_input.py

18

Extra Credit Opportunities

- Prefer you do extra credit on labs, but...
- If you read any good articles over break and summarize them on the blog, you can earn extra credit points
 - The top 5 most interesting articles/summaries get extra points!

Nov 16, 2007

Sprenkle - CS111

19

Broader Issue

- Facebook's News Feed
- Discussion:
 - What are the pros and cons of the News Feed?
 - What are Facebook's privacy and security issues?
 - How does Facebook address these issues?
 - Why has Facebook been so much more successful than predecessors such as Friendster and Orkut?

Nov 16, 2007

Sprenkle - CS111

20

Discussion

- Good algorithm → Business success
 - Google's PageRank algorithm
 - Facebook's Newsfeed algorithm
 - Others?
- Relevance to class:
 - Algorithm uses frequency tables (which we've used a lot!) but fancier
- Be careful with Facebook (and MySpace and others) when you're job hunting

Nov 16, 2007

Sprenkle - CS111

21