

## Objectives

- Security Considerations
- What is Computer Science?
- Final Exam questions

Apr 4, 2012

Sprenkle - CS111

1

## Questions about Lab 11?

Apr 4, 2012

Sprenkle - CS111

2

## Lab11 Search Review

- What does your implemented binary search algorithm become in “the worst case”?
- Assumption: only a few people in a given network
  - Otherwise, maybe just as well to use linear search

Apr 4, 2012

Sprenkle - CS111

3

## SECURITY CONSIDERATIONS

Apr 4, 2012

Sprenkle - CS111

4

## Security Considerations

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

Apr 4, 2012

Sprenkle - CS111

5

## Simple Example of Security Bug

- Example from a real web site
  - Simulated here
- More fundamental than a security bug
  - Hopefully, CS111 students would know better
    - May not always check user input but know paid programmers should do it!
- How to address this security issue?

Apr 4, 2012

Sprenkle - CS111

6

## Another Example

- Modifying URL parameters

```
http://network.domain.com/login.jsp?room=4531&rate=9.99
```

```
http://network.domain.com/login.jsp?room=4531&rate=0.00
```

Learn more in CSCI335!

Apr 4, 2012

Sprenkle - CS111

7

## Security Bug in Python

- Demonstrate with `security_bug.py`

Apr 4, 2012

Sprenkle - CS111

8

## Why the Security Bug?

- `eval(input(prompt))`
- Example:
  - `val = eval(input("Enter a value: "))`
  - In terminal:
    - Enter a value: `1 + 5 + 6`
  - `val` is assigned `12`
- Example:
  - `total += eval(input("Enter deposit: "))`
  - Enter deposit: `total * 3`
  - `total` is assigned `total+total*3`

Apr 4, 2012

Sprenkle - CS111

9

## Handling the Security Bug

- Cast `input` to an integer

```
user_input = input("Enter a deposit")  
total += int(user_input)
```

- Casting to an `int` will throw an exception if we try to input `total`
- Use `try/except` to handle error

Apr 4, 2012

Sprenkle - CS111

`fixed_input.py`

10

## More on `input` function

- Explains the error message we got if we tried to input `B6`
  - Looks for variable named `B6`

```
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
```

Apr 4, 2012

Sprenkle - CS111

11

## 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?

Apr 4, 2012

Sprenkle - CS111

12

## 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 careless

How to address issues?

Apr 4, 2012

13

## 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

Apr 4, 2012

Sprenkle - CS111

14

## File Input from User

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

Apr 4, 2012

Sprenkle - CS111

file\_input.py

15

## Handling Security

- **Prevention**
  - Don't allow bad stuff
  - Examples: validate input, restrict input files
- **Detection**
  - After see bad, stop it
  - Example: Microsoft's Halo
  - Example: network usage

Apr 4, 2012

Sprenkle - CS111

16

## COMPLEXITY SCIENCE

Apr 4, 2012

Sprenkle - CS111

17

## CS == Complexity Science

- How can it be done?
  - Based on **information**
  - Managing, manipulating data
  - Possible algorithms
- How well can it be done?
  - Most **efficient** algorithm in terms of time and/or space
- Can it be done at all?
  - Often, proof is a program--an implementation of the above

Jan 9, 2012

Sprenkle - CSCI 111

18

## Computer Science != Programming

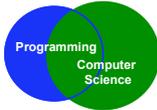
programming : CS ::

machining : engineering

grammar : literature

equations : mathematics

walking : W&L



a vehicle, not a destination

Apr 4, 2012

Sprenkle - CS111

19

## Computer Science Fields

### Systems

- Architecture
- Operating systems
- Networks
- Distributed and parallel systems
- Databases
- Security
- ...

### Software

- Compilers
- Graphics
- Software engineering
- Software testing and verification
- ...

### Theory

- Algorithms
- Theory of computation
- ...

### Other

- Artificial intelligence
- Robotics
- Natural language processing
- Bioinformatics
- Visualization
- Numerical analysis
- ...

- Often research involves combinations of these fields
- Not just programming!
  - But programming is a tool to do much, much more!

Apr 4, 2012

Sprenkle - CS111

20

## Computer Science Fields

### Systems

- Architecture \*
- Operating systems \*
- Networks \*
- Distributed \* and parallel systems
- Databases \*
- Security \*
- ...

### Software

- Compilers
- Graphics \*
- Software engineering \*
- Software testing \* and verification \*
- ...

### Theory

- Algorithms \*
- Theory of computation
- ...

### Other

- Artificial intelligence \*
- Robotics \*
- Natural language processing \*
- Bioinformatics
- Visualization \*
- Numerical analysis
- ...

\* = field we discussed or did a problem in

➢ Some are a stretch :)

Apr 4, 2012

Sprenkle - CS111

21

## Final Exam Take Home Questions

- 2 essay questions about the Broader Issues
- **Due before end of exam period**
  - 5 p.m. Friday
- Each essay should be about 1/2 a page
- Goal: answer the question clearly, precisely, and convincingly
  - Not too wordy
  - Evidence/examples to support your argument
  - Correct spelling, grammar, punctuation

Apr 4, 2012

Sprenkle - CS111

22

## Final Exam Review

- Focus on object-oriented programming
- New content: 2D lists, exceptions, security issues, complexity science
- Cumulative:
  - Functions, data types, common methods & operations
  - How to model data

Apr 4, 2012

Sprenkle - CS111

23

## Final Exam Review

- What do you need to do to be able to use methods from a class?
- What are the different ways to iterate through a list?
- How can you iterate through a dictionary?

Apr 4, 2012

Sprenkle - CS111

24

## Reminders

- Broader Issue: One Laptop Per Child
- Final exam envelopes!
- Finish Lab 11
- Study for Exam
- Do ConnectFour extra credit (additional 2D practice)
- Course Evaluations
  - [Sakai – tests & quizzes](#)