

Objectives

- Review parts of algorithms
- More on working with text files
- Intro to Lists
- Broader Issues in Computer Science

Oct 19, 2007

Sprenkle - CS111

1

Parts of an Algorithm

- Primitive operations
 - What data you have, what you can do to the data
- Naming
 - Identify things we're using
- Sequence of operations
- Conditionals
 - Handle special cases
- Repetition/Loops
- Subroutines
 - Call, reuse similar techniques

An overview for the semester!

Have we covered all these?
How do we implement them in Python?

Oct 19, 2007

Sprenkle - CS111

2

Parts of an Algorithm

- Primitive operations
 - What data you have, what you can do to the data
- Naming
 - Identify things we're using
- Sequence of operations
- Conditionals
 - Handle special cases
- Repetition/Loops
- Subroutines
 - Call, reuse similar techniques

Here is where most of the rest of the semester focuses
No longer primitive

An overview for the semester!

Have we covered all these?
How do we implement them in Python?

Oct 19, 2007

Sprenkle - CS111

3

Object-Oriented Programming

- **Classes** (or **types**) define the **methods** that you can perform on **objects** of that class/type
- Methods are similar to functions but **called/used** differently:
 - `objectname.methodname([parameters])`
 - Examples of using **string** methods:
`animal.upper()`, `sound.center(10)`

Oct 19, 2007

Sprenkle - CS111

4

Files

- Conceptually, a file is a **sequence** of data stored in memory
- To use a file in a Python script, create an object of type **file**
 - `<varname> = file(<filename>, <mode>)`
 - `<filename>` : string
 - `<mode>` : string, either "r" for read or "w" for write
 - Example: `dataFile = file("years.dat", "r")`

Known as the **constructor** - "constructs" a file object

Oct 19, 2007

Sprenkle - CS111

5

Reading from a File

- `read()` - entire contents into a string
- `readline()` - one line from file into a string
- **for** loop
 - Read as: for each line in the file, do something

A line (of type **string**) from the file
file object
for line **in** dataFile:
 print line

Oct 19, 2007

Sprenkle - CS111

6

Problem: Ignore Comments

- Twist on problem from last time
 - Assumed that comments were at the beginning of the line
- Ignore everything after the '#'
 - Or look for the term in everything before the '#'

Oct 19, 2007

Sprenkle - CS111

7

Handling Numeric Data

- We have been dealing with reading and writing strings so far
- What do we need to do to read **numbers** from a file?
 - Cast as a numeric type, e.g., int or float
- How can we write numbers to a file?
 - Cast number as a string

Oct 19, 2007

Sprenkle - CS111

8

Problem: Temperature Data

- You have data files that contain the daily high temperatures for the last year for several locations. What is the average high temperature (to 2 decimal places) for the location?
 - The data file contains one temperature per line
 - Example: data/florida.dat
- Rule of Thumb: Always look at the data file before you try to process it

Oct 19, 2007

Sprenkle - CS111

9

Problem: Create a Summary Report

- We have a file containing students names and their years (freshman, sophomore, junior, or senior) for this class. We want to create a report (in a file) that says the year and how many students from that year are in this class.
 - Again, we want to ignore comments in the file

Do we need to start this program from scratch or do we have some code that we can use or "repackage"?

Oct 19, 2007

Sprenkle - CS111

10

Other Sequences of Data

- We commonly group a sequence of data together and refer to them by one name
 - Days of the week: Sunday, Monday, Tuesday, ...
 - Months of the year: January, February, March, ...
 - Shopping list
- Can represent this data as a **list** in Python
 - Similar to **arrays** in other languages

Oct 19, 2007

Sprenkle - CS111

11

Examples of Lists in Python

- daysInWeek=["Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"]
- monthsInYear=["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"]
- groceryList=["milk", "eggs", "bread", "Doritos", "OJ", "sugar"]

Oct 19, 2007

Sprenkle - CS111

12

Benefits of Lists

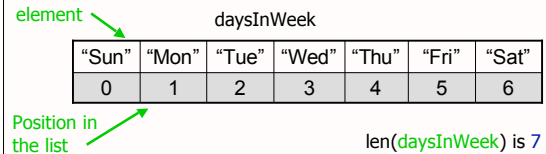
- Group related items together
 - Instead of creating separate variables
 - sunday = "Sun"
 - monday = "Mon"
- Convenient for dealing with large amounts of data
 - Example: could keep all the temperature data in a list if needed to reuse later
- Functions and methods for handling, manipulating lists

Oct 19, 2007

Sprenkle - CS111

13

Lists: A Closer Look



- `<listname>[<int_expr>]`
 - Similar to accessing characters in a string
 - `daysInWeek[-1]` is "Sat"
 - `daysInWeek[0]` is "Sun"

Oct 19, 2007

Sprenkle - CS111

14

List Operations

Concatenation	<code><seq> + <seq></code>
Repetition	<code><seq> * <int_expr></code>
Indexing	<code><seq>[<int_expr>]</code>
Length	<code>len(<seq>)</code>
Slicing	<code><seq>[:]</code>
Iteration	<code>for <var> in <seq>:</code>
Membership	<code><expr> in <seq></code>

Similar to operations for strings

Oct 19, 2007

Sprenkle - CS111

15

Iterating through a List

- Read as
 - For every element in the list ...
-
- Equivalent to
 - `for x in xrange(len(list)):` *Iterates through positions in list*
`print list[x]`

Oct 19, 2007

Sprenkle - CS111

daysOfWeek.py

16

Broader Issues in Computer Science

- Testing isn't a broader issue
 - Glad you noticed lots of the issues with testing
 - We'll keep talking about it because I love it!
- Is the Excel 2007 a "reasonable" bug?
 - Why wasn't it caught?
 - Should it have been caught?

Groups: Freshmen
Juniors/Seniors

Oct 19, 2007

Sprenkle - CS111

17

Broader Issues in Computer Science

- Have you ever encountered a bug in a program?
 - What happened?
 - How severe was the problem? Were you able to recover?
 - How did you respond? (Angry? Didn't think about? ...)
- If people can recover from a bug, when does it become important for software developers to fix the problem?
 - Tradeoffs between costs/revenues of implementing new features versus fixing existing code
 - What matters to you (as a consumer) more?

Oct 19, 2007

Sprenkle - CS111

18

Notes from a Keynote Speech about Testing Microsoft Vista

- Users are “trained” to not use buggy features
 - After user encounters a certain bug when doing something enough times, eventually, the user stops trying to do that

$$\text{User's Loss in Confidence} = \text{Disruption Frequency} \times \begin{matrix} \text{Recovery Time} \\ \text{Recover Effort} \\ \text{Lost data} \\ \text{Uncertainty} \end{matrix}$$

- Only ship fixes that affect many users

Oct 19, 2007

Sprenkle - CS111

19

Status from Official Excel Blog

- Post on 9/25 Happy Ending
 - We've come up with a fix for this issue and are in the final phases of a broad test pass in order to ensure that the fix works and **doesn't introduce any additional issues** - especially any other calculation issues. This fix then needs to make its way through our official build lab and onto a download site - which we expect to happen very soon.
- Post on 10/9
 - As of today, fixes for this issue in Excel 2007 and Excel Services 2007 are available for download ...
 - We are in the process of adding this fix to Microsoft Update so that it will get **automatically pushed** to users running Excel 2007 or Excel Services 2007. Additionally, the fix will also be contained in the first service pack of Office 2007 when it is released (the release date for SP1 of Office 2007 has not been finalized).

Oct 19, 2007

Sprenkle - CS111

20

Problem: Music Files

- We have an album file that has the format
 - <Artist name>
 - <Album name>
 - <Song name 1>
 - <Song length 1>
 - ...
 - <Song name n>
 - <Song length n>Length has the format min:seconds
- Given an album file, print out the number of songs and the total length of the album

Oct 19, 2007

Sprenkle - CS111

21