Announcements for Today

Assignment 1
- We have finished grading! Resubmit until correct
- If you were close...
  - Will get feedback in CMS
  - Fix your assignment
- If you were very wrong...
  - You got an e-mail
  - Holding 1-on-1s this week
- FINISH THE SURVEY

Reading
- Read Chapter 4
- No reading for Thursday

More Assignments
- Please reread A2 instructions
  - Major typos fixed in rot2_y
  - A3 posted on Monday
  - Due 2 weeks from Fri

Algorithms: Heart of Computer Science
- Algorithm: A step-by-step procedure for how to do something (usually a calculation).
- Implementation: How to write an algorithm in a specific programming language
- Good programmers know how to separate the two
  - Work out algorithm on paper or in head
  - Once done, implement it in the language
  - Limits errors to syntax errors (easy to find), not conceptual errors (much, much harder to find)
- Key to designing algorithms: stepwise refinement

Stepwise Refinement: Basic Principles
- Write Specifications First
  - Write a method specification before writing its body
- Take Small Steps
  - Do a little at a time; follow the Mañana Principle
- Run as Often as You Can
  - This can catch syntax errors
- Separate Concerns
  - Focus on one step at a time
- Intersperse Programming and Testing
  - When you finish a step, test it immediately

Mañana Principle
- If not in current step, delay to “tomorrow”
  - Use comments to write steps in English
  - Add “stubs” to allow you to run program often
  - Slowly replace stubs/comments with real code
  - Only create new local variables if you have to
  - Sometimes results in creation of more functions
    - Replace the step with a function call
    - But leave the function definition empty for now
  - This is called top-down design

Function Stubs

<table>
<thead>
<tr>
<th>Procedure Stubs</th>
<th>Fruitful Stubs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single statement: pass</td>
<td>Single return statement</td>
</tr>
<tr>
<td>Body cannot be empty</td>
<td>* Type should match spec.</td>
</tr>
<tr>
<td>This command does nothing</td>
<td>* Return a “default value”</td>
</tr>
</tbody>
</table>
| Example: def foo(): pass | Example: def first_four_letters(s):
| | return '' # empty string |

Purpose of Stubs
Create a program that may not be correct, but does not crash.

Example: Reordering a String

```python
def last_name_first(s):
    """Returns: copy of s in form <lastname>, <firstname>
    Precondition: s is in the form <firstname> <lastname>
    with one blank between the two names"
    # Find the first name
    # Find the last name
    # Put them together with a comma
    return '' # Currently a stub
```

last_name_first('Walker White') is 'White, Walker'
Example: Reordering a String

Example: Reordering a String

Example: Reordering a String

Example: Reordering a String

Refinement: Creating Helper Functions

Refinement: Creating Helper Functions

Exercise: Anglicizing an Integer

Exercise: Anglicizing an Integer

Exercise: Anglicizing an Integer

Exercise: Anglicizing an Integer

Exercise: Anglicizing an Integer