Objectives

• Concluding CS111
  ➢ Other programming languages
  ➢ What is computer science?

Applying What You Know To Other Languages

• At the beginning of the semester, some of you asked
  ➢ “Why Python?”
  ➢ “Will I be able to read/write programs in other programming languages?”
• We’ll answer the first by showing that you can do the second

Applying What You Know To Other Languages

• Syntax: symbols used
• Semantics: what the symbols mean

What is the Python Program Doing?

• Page 4 of handouts

What is the Python Program Doing?

• Getting a line of input from “standard in” (from the user)
• Splitting the input into integers
• Calculating the result to a formula
• Deciding if a student is admitted, based on the result of the formula

Admissions Problem

• Binary University decides to admit students based on a formula that weighs various factors
  ➢ Scores of 70 or better are admitted
• Input: single line, 4 integers, in order below

<table>
<thead>
<tr>
<th>Category</th>
<th>Range</th>
<th>Weight Factor (Multiplier)</th>
</tr>
</thead>
<tbody>
<tr>
<td>High School GPA</td>
<td>0 - 10</td>
<td>0.25</td>
</tr>
<tr>
<td>SAT score</td>
<td>600-2400</td>
<td>.01</td>
</tr>
<tr>
<td>AP Courses</td>
<td>0-10</td>
<td>10</td>
</tr>
<tr>
<td>Intangibles</td>
<td>1-10</td>
<td>8</td>
</tr>
</tbody>
</table>
Example Input/Expected Output

<table>
<thead>
<tr>
<th>Input</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 0 300</td>
<td>DENY</td>
</tr>
<tr>
<td>6 10 99 2390</td>
<td>ADMIT</td>
</tr>
<tr>
<td>0 7 82 1500</td>
<td>ADMIT</td>
</tr>
<tr>
<td>2 5 0 990</td>
<td>DENY</td>
</tr>
<tr>
<td>2 5 0 1000</td>
<td>ADMIT</td>
</tr>
<tr>
<td>2 5 0 1010</td>
<td>ADMIT</td>
</tr>
</tbody>
</table>

What is the Python Program Doing?

- Getting a line of input from “standard in” (from the user)
- Splitting the input into integers
- Calculating the result to a formula
- Deciding if a student is admitted, based on the result of the formula

Identify these pieces in the other programs

Example Programs

- printLab.sh
  - Bash script

Comparing Programming Languages

- How is the syntax/semantics of these languages different from Python?
  - What is easier or harder to do in these other programming languages than in Python?

Comparing Programming Languages

- Benefits of Python:
  - Simpler syntax (e.g., fewer {} and ())
  - Can cover some content with less overhead
- Drawbacks
  - Data types aren’t explicit (static)
  - Can be harder for you to remember and keep straight
  - Not compiled explicitly beforehand
  - Keep executing to find all the syntax bugs
  - Doesn’t check: “you’re passing a file instead of a string”
  - Allows you to do some crazy stuff that won’t work in other programming languages

Who Uses Python?

- Google
  - Backends of Gmail and Google Maps and search-engine internals
- NASA
  - Collaborative engineering
- Yahoo
  - Groups: Maintain discussion groups; Maps
- RedHat Linux
  - System infrastructure
- Original BitTorrent client; Youtube; Civilization IV
### Computer Science Fields

- **Systems**
  - Architecture
  - Operating systems
  - Networks
  - Distributed and parallel systems
  - Databases
  - Security

- **Software**
  - Compilers
  - Graphics
  - Software engineering
  - Software testing

- **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!

---

### CS == Complexity Science

- **Study of Complexity**
  - 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

---

### Broader Issues

- **Articles:**
  - Tech education, Puzzles of Cyberspace, DARPA Urban challenge, Excel Bug, Metaphors, Sensor Networks, Social Networks, OLPC
- **Questions**
  - Most liked article? Least liked article?
  - Who found the articles overall least interesting?
  - Most interesting?

- **Group 1:** Russ, Carrie, Mallory, Chen, Ben
- **Group 2:** Craig, Aaron, Kevin, Sara, Dylan
- **Group 3:** Thomas, David, Charles, Michelle
- **Group 4:** Greg, Camille, Mike, Taylor

- **One Laptop Per Child**
  - An experiment on bringing cheap but educational technology to poor children
- **What challenges did OLPC face and how did that affect their design decisions?**
- **What are some unusual features of the laptop?**
- **What does this technology mean for better-off countries?**
- **Is this project worthwhile?**
### Discussion

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Design Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of power</td>
<td>New, cheap battery; Consumes less power; Alternative power sources: solar power, pull cord</td>
</tr>
<tr>
<td>Software bloat</td>
<td>Rewrite code more compactly, efficiently</td>
</tr>
<tr>
<td>Environment</td>
<td>Dust proof, drop proof, light</td>
</tr>
<tr>
<td>Users: children</td>
<td>Simple user interfaces; tiny keyboard; lightweight; applications keep students interested</td>
</tr>
<tr>
<td>Cost</td>
<td>Linux, Python, open-sources tools; cheaper battery; no hard drive; no CD/DVD drive</td>
</tr>
</tbody>
</table>

### Conclusions

- See impact of computer science on your life
- Understand some of the computing issues better
  - Taking out some of the mystery
  - Security, testing, debugging, efficiency
- Algorithms are everywhere
  - Process for solving problems
  - Mapping human intuition to systematic/automatic process