CS 160
CS Orientation

Data Types, Conditionals, and Repetition
Hierarchy of Operations

- Functions, i.e. math.sqrt()
- Power
- Mod
- Mult, Div
- Add, Sub
- Relational
- Logical

- Binary operators (2 operands)
- Unary operators (1 operand)
- Parentheses

Happy 4 operators!
Python Examples

- $5 \times 2 + 3 - 10$
- $5 \times (2 + 3) - 10$
- $3 > 2 + 4$
- Etc.
Data Types/Converting

- string - “string of text”
- character – ‘a’
- integer – 79
- boolean – True
- float – 79.0
How about storing values?

- \( \text{ch} = 'a' \)
- \( \text{num} = 79 \)
- \( \text{cont} = \text{True} \)
- \( \text{num} = \text{num} + 1 \)
- Can we mix types?
  - \( \text{num} = \text{num} + \text{cont} \)
  - \( \text{num} = \text{num} + \text{ch} \)
  - \( \text{ch} = \text{ch} + \text{num} \)
Python Demo...

```
access.engr.oregonstate.edu - PuTTY

shinnje  pts/153  50.90.178.82  13:28  3.00s  0.16s  0.09s  vim numGuess.cpp
fritterj pts/154  98.232.169.70  13:33  24:59  0.06s  0.06s  -tcsh
 carrilca pts/155  50.205.154.99  13:33  43:00s  0.06s  0.06s  -bash
 kodys   pts/156  73.237.228.170  13:33  3.00s  0.44s  0.39s  vim minmax.cpp
 chickj  pts/157  10.251.86.68   13:35  11:31  0.09s  0.09s  -tcsh
 rimerr  pts/158  10.248.186.89  13:41  3.00s  0.10s  0.04s  vim stats
 pelletim pts/178  72.0.183.173  Tue20  46:51  0.11s  0.03s  vim Box.hpp

flip2 ~ 151% cd cs160/private/
flip2 ~/cs160/private 152% python3
Python 3.3.2 (default, May 10 2014, 10:24:41)
[GCC 4.8.2 20140120 (Red Hat 4.8.2-16)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> num=67
>>> cont=True
>>> ch='e'
>>> num=num+1
>>> print(num)
68
>>> num=num
>>> print(num)
136
>>> num+=1
>>> print(num)
137
```
Python Demo...

```
136 >>> num+=1
137 >>> print(num)
138 >>> print(num+cont)
139 >>> print(num+ch)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for +=: 'int' and 'str'
>>> ord(ch)
101
>>> print(num+ord(ch))
238
>>> print(num+int(ch))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() with base 10: 'e'
>>> ch='2'
>>> print(num+int(ch))
139
>>> print(chr(56))
8
```
## Sequential Logic Structure

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Flowchart</th>
<th>Pseudocode</th>
</tr>
</thead>
<tbody>
<tr>
<td>5. Instruction</td>
<td><img src="image" alt="Flowchart Diagram" /></td>
<td>Instruction</td>
</tr>
<tr>
<td>6. Instruction</td>
<td></td>
<td>Instruction</td>
</tr>
<tr>
<td>7. Instruction</td>
<td></td>
<td>Instruction</td>
</tr>
<tr>
<td>8. :</td>
<td></td>
<td>Instruction</td>
</tr>
</tbody>
</table>

*Oregon State University*
Python Sequential Logic

print("   *   ")
print("  ***  ")
print(" ***** ")
print("*******")
## Decision Logic Structure

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Flowchart</th>
<th>Pseudocode</th>
</tr>
</thead>
<tbody>
<tr>
<td>::</td>
<td><img src="image" alt="Decision Logic Diagram" /></td>
<td>::</td>
</tr>
<tr>
<td>5. If <code>&lt;decision&gt;</code> then Instruction else Instruction</td>
<td></td>
<td>If <code>&lt;decision&gt;</code> then Instruction else Instruction Endif</td>
</tr>
<tr>
<td>6. ::</td>
<td></td>
<td>::</td>
</tr>
</tbody>
</table>
Relational Operators and Symbols

- >
- >=
- <
- <=
- ==
- !=

binary

2 operands

equal to
not equal to
Logical Operators and Symbols

- **not**
  - Binary: $\neg T \rightarrow F$
  - Ternary: $\neg F \rightarrow T$

- **and**
  - Short circuit: $T \land T = T$
  - $T \land F = F$
  - $F \land F = F$

- **or**
  - Short circuit: $T \lor T = T$
  - $T \lor F = T$
  - $F \lor F = F$

True or False