CS 160
CS Orientation

Expressions, Data Types, & Input/Output
Chap. 2
## Expressions and Equations

<table>
<thead>
<tr>
<th>Expressions</th>
<th>Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A + B )</td>
<td>( C = A + B )</td>
</tr>
<tr>
<td><em>A</em> and <em>B</em> are numeric.</td>
<td><em>C</em>, <em>A</em>, and <em>B</em> are numeric.</td>
</tr>
<tr>
<td>The resultant is numeric and</td>
<td>The resultant is stored in</td>
</tr>
<tr>
<td>is not stored.</td>
<td><em>C</em>.</td>
</tr>
<tr>
<td>( A &lt; B )</td>
<td>( C = A &lt; B )</td>
</tr>
<tr>
<td><em>A</em> and <em>B</em> are numeric,</td>
<td><em>A</em> and <em>B</em> are numeric,</td>
</tr>
<tr>
<td>character, or string.</td>
<td>character, or string.</td>
</tr>
<tr>
<td>The resultant is logical and</td>
<td>The resultant is stored in</td>
</tr>
<tr>
<td>is not stored.</td>
<td><em>C</em>; <em>C</em> is logical.</td>
</tr>
<tr>
<td>( A \ OR \ B )</td>
<td>( C = A \ OR \ B )</td>
</tr>
<tr>
<td><em>A</em> and <em>B</em> are logical.</td>
<td><em>C</em>, <em>A</em>, and <em>B</em> are logical.</td>
</tr>
<tr>
<td>The resultant is logical and</td>
<td>The resultant is stored in</td>
</tr>
<tr>
<td>is not stored.</td>
<td><em>C</em>.</td>
</tr>
</tbody>
</table>
Operators and Their Computer Symbols

- addition (+)
- subtraction (-)
- multiplication (*)
- division (/)
- integer division (//)
- modulus (%) 
- exponentiation (**)

(concatenation of strings)

Examples:
2 + 3 = 5
2 - 3 = -1
2 * 3 = 6
2 / 3 = 0.6666666666666666
2 // 3 = 0
2 % 3 = 2
2 ** 3 = 8

# comment
Relational Operators and Symbols

return a boolean (True/False)

• >  - greater than
• >= - gr than equal to
• <
• <=
• ==  \rightarrow equal to
• !=  \rightarrow no equal to
Logical Operators and Symbols

- **not**
- **and**
- **or**

- Both have to be true to be true
- Either is true and will be true
- True or False
Hierarchy of Operations

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

- $5 \times 2 + 3 - 10$
- $5 \times (2 + 3) - 10$
- `not True or False`
- $3 > 2 + 4$
- `True and True or True and False`
- `((True and True) or True) and False`
- `not 3 < 2 and True or False`
Python Demo/Expressions

```python
>>> 5 * 2 + 3 - 10
3
>>> 5 * (2 + 3) - 10
15
>>> not True or False
False
>>> 3 > 2 + 4
False
>>> True and True or True and False
True
>>> (3 > 2) + 4
5
>>> (True and True) or (True and False)
True
>>> ((True and True) or True) and False
False
>>> not 3 < 2 and True or False
True
>>> |
```
Data Types/Converting

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

- ch = 'a'
- num = 79
- cont = True
- num = num + 1

Can we mix types?
- num = num + continue
- num = num + ch
- ch = ch + num

- num = 80
- ch = 'b'
- continue = 1

- can change the types
Python Demo/Expressions

```python
Type "copyright", "credits" or "license()" for more information.
>>> ch = 'a'
>>> print(ch)
a
>>> num = 79
>>> print(num)
79
>>> cont = True
>>> print(cont)
True
>>> num = num + 1
>>> print(num)
80
>>> num = num + cont
>>> print(num)
81
>>> num = num + ch
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    num = num + ch
TypeError: unsupported operand type(s) for +: 'int' and 'str'
>>> num = num + ord(ch)
>>> print(num)
178
>>> ch = ch + num
Traceback (most recent call last):
  File "<pyshell#13>", line 1, in <module>
    ch = ch + num
TypeError: Can't convert 'int' object to str implicitly
>>> ch = ch + chr(num)
>>> print(ch)
a
>>> 
```