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
Python Examples

• 5 * 2 + 3 - 10
• 5 * (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?

• `ch = ‘a’`
• `num = 79`
• `cont = True`
• `num = num + 1`
• **Can we mix types?**
  – `num = num + continue`
  – `num = num + ch`
  – `ch = ch + num`
Python Data Type Demo...
Sequential Logic Structure

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Flowchart</th>
<th>Pseudocode</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>5. Instruction</td>
<td>Instruction</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>...</td>
</tr>
</tbody>
</table>

Instruction

Instruction

Instruction

Instruction
Python Sequential Logic

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

<table>
<thead>
<tr>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
</tr>
<tr>
<td>5. If <code>&lt;decision&gt;</code> then Instruction else Instruction</td>
</tr>
<tr>
<td>6. ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flowchart</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Flowchart Diagram" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pseudocode</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
</tr>
<tr>
<td>If <code>&lt;decision&gt;</code> then Instruction else Instruction Endif</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>
Relational Operators and Symbols

- >
- >=
- <
- <=
- ==
- !=
Logical Operators and Symbols

• not
• and
• or
Python Decision Logic:
Print 1, 3, 5, or 7 stars
Differences/Similarities in these?

```python
x=int(input(“Print 1, 3, 5, 7 stars?”));

if(x==1):
    print("   *   ")
elif(x==3):
    print("  ***  ")
elif(x==5):
    print(" ***** ")
elif(x==7):
    print("*******")
```

```python
if(x==1):
    print("   *   ")
elif(x==3):
    print("  ***  ")
elif(x==5):
    print(" ***** ")
elif(x==7):
    print("*******")
```
Python Decision Logic:
Print 1, 3, 5, or 7 (for any other #) stars
Differences/Similarities in these?

```python
x=int(input("Print 1, 3, 5, 7 stars?")));  

if(x==1):
    print("   *   ")
if(x==3):
    print("  ***  ")
if(x==5):
    print(" ***** ")
else:
    print("*******")
```

```python
if(x==1):
    print("   *   ")
elif(x==3):
    print("  ***  ")
elif(x==5):
    print(" ***** ")
else:
    print("*******")
```
Python Conditional Demo...
Loop Logic Structure

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Flowchart</th>
<th>Pseudocode</th>
</tr>
</thead>
<tbody>
<tr>
<td>5. Loop</td>
<td><img src="image" alt="Flowchart Diagram" /></td>
<td>Loop</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Instruction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Instruction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Instruction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Until &lt;logical expression&gt;</td>
</tr>
<tr>
<td>6.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
Loop
Instruction
Instruction
Instruction
Until <logical expression>
```
Python Loop Logic

for x in range(7):
    print("*", end="")

OR

x=1
while(x<=7):
    print("*", end="")
    x+=1
In-class Exercise #2

• Explain these terms:
  Conditional execution, repetition, for versus while loop

• Explain what each of line in this code does and tell me what the output is:

```python
y=0;
for x in range(10):
    if(x%2):
        y=y+1;
print(y);
```

• Write a loop to print the following picture:

```python
print("   *");
print("  ***");
print(" *****");
print("*******");
```
Python **While** Demo...

- While the user wants to continue to enter a number for stars to print...