CS 160
CS Orientation

Input/Output, Conditionals, and Loops

Oregon State University

Relational Operators and Symbols

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

```
if ($ >= 20)
    eat at Flat Tails
else if ($ >= 10)
    eat at subway
else
    stay home + eat
```

Logical Operators and Symbols

- not
- and
- or

not F T

not T F

T and T T

T and F F

short circuit { F and T F

F and F F

short circuit { T or T T

T or F T

F or T T

F or F F

Python Examples

- 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

use parens for precedence

Python Decision Logic: Print 1, 3, 5, or 7 stars

Differences/Similarities in these?

same output

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

more expensive w/ 1, 3, 5

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

```
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?

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

5 & any other number give same results

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

1 & 3 print respective stars & 7 stars

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

any num other than 1,3,5 prints 7

Exercise

- Write an algorithm that will tell a user whether they have entered a valid triangle using the triangle inequality property (any sum of 2 sides cannot be less than the third side).

6. ENGR

Re-attach Fullscreen Stay on top Duplicate Close

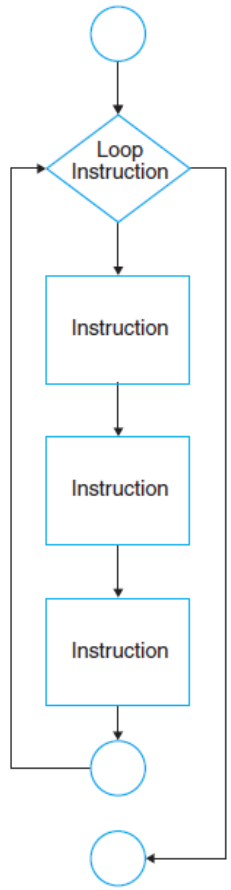
```
#prompt user for side 1
#read side1 value
#prompt user for side 2
#read side2 value
#prompt user for side 3
#read side3 value

#if(side1+side2 < side3)
   #print not a triangle
#else if(side1+side3 < side2)
   #print not a triangle
#else if(side2+side3 < side1)
   #print not a triangle
#else
   #print a good triangle


#if((side1+side2 < side3) or (side2+side3 < side1) or (side1+side3 < side2))
   #print not a triangle
#else
   #print a good triangle

~
~
~
                                                    7,0-1          All
```

Loop Logic Structure

Algorithm	Flowchart	Pseudocode
⋮ 5. Loop Instruction Instruction Instruction Until <logical expression> 6. ⋮		⋮ 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
```

Exercise

- How about if we alter this to allow a user to do this for any number of triangles?

① maybe ask user how many triangle they want to check + for each triangle, ask sides + check if good

② while the user wants to check a triangle, ask for sides + check if good