

# CSC 110 Week 5, Lecture 2

Dr. Bodine

# Last week...

- Functions
  - Subprograms
  - Modular design
- Parameters
  - Inputs to functions
  - Modifying parameters....
- Returns
  - Outputs from functions
- Scope!!!!

## Next week...

- Midterm on Tuesday:
  - 1 hour long
  - Administered on Canvas
  - Chapters 1-3, 5-7 of Zelle
- Combination of quiz-like questions and short programming questions
  - Example: write a short script that opens a new file called test.txt and write "Hello, World!" into the file.

# This week

- Decision structures for basic control
  - Logic!
- If statements!!!
  - Yay! For everyone who has wanted to use them on the homework
  - If, If-else, if-elif-else
- Boolean logic
  - Two states: True or False, 1 or 0.
- Exception handling

# Basic decisions

- Differentiate action based on result of a simple true-false questions
  - Example: If student got >89, print that student got an A.
- Alternative flow of the program
  - If condition is met, code in loop executes
  - If condition is not met, code in the loop does not execute.



# Writing if statements in python

```
if <condition>:  
    <body>
```

- **Example:**

```
if x<0:  
    print("x is negative!")
```

# Basic conditions

- Two expressions and a relational operator.
- Example:
  - $x < 0$
  - $x-1 < 0$
  - $x-y > 100$
  - $\text{math.exp}(x) \leq 100$



# Condition symbols

$==$  “is equal to”

$>$  “is greater than”

$<$  “is less than”

$>=$  “is greater than or equal to”

$<=$  “is less than or equal to”

$!=$  “is NOT equal to”

## Discuss with your neighbors...

- Which of the following are valid conditions in python? Assume x and y have been assigned values.
  - $x \geq 3.1415$
  - $x + 7 < 9$
  - $x+7 \neq 42$
  - $x+y \leq 100$
  - $x+y = 100$
- Which of the valid conditions are True for  $x = 2$  and  $y = 10$ ?

# Boolean logic

- Boolean expressions
  - Statements are either **True** or **False**
  - Languages code them differently: 1 and 0; True and False; true and false; ...
- Can build up more complicated conditions
  - Chain together boolean statements
  - AND
  - OR
  - XOR

## Boolean logic (cont...)

- And: True if both statements must be true
- Or: True if *either* statement is true
- Exclusive or: True if *only one* of the two statements is true
  - Often called xor

## Boolean logic (cont...)

P	Q	P and Q	P or Q	P xor Q
True	True			
True	False			
False	True			
False	False			

## Boolean logic (cont...)

P	Q	P and Q	P or Q	P xor Q
True	True	True	True	False
True	False	False	True	True
False	True	False	True	True
False	False	False	False	False



# Writing if statements in python

```
if <condition>:  
    <body>
```

- **Example:**

```
if x<0:  
    print("x is negative!")
```



# Example if statement

```
#Test script for week 5, lecture 1

def main():
    speed = eval(input("Please enter how fast you were going in mph: "))

    if speed>35:
        print("You were going over the speed limit! Please slow down!")

main()
```

## Example if statement (cont...)

```
#Test script for week 5, lecture 1

def main():
    speed = eval(input("Please enter how fast you were going in mph: "))

    if speed>35:
        print("You were going over the speed limit! Please slow down!")

main()
```

```
>>> import wk5l1
Please enter how fast you were going in mph: 50
You were going over the speed limit! Please slow down!
>>> wk5l1.main()
Please enter how fast you were going in mph: 35
>>>
```

# Adding more options

- else statements
  - Says what to do if the condition is not met
- elif (else if) statements
  - Says to check something else if the initial condition is not met
  - Can be stacked...create many, many different paths

# else example

```
def main():  
    speed = eval(input("Please enter how fast you were going in mph: "))  
  
    if speed>35:  
        print("You were going over the speed limit! Please slow down!")  
    else:  
        print("Thank you for obeying the speed limit.")  
  
main()
```

## else example (continued...)

```
def main():
    speed = eval(input("Please enter how fast you were going in mph: "))

    if speed>35:
        print("You were going over the speed limit! Please slow down!")
    else:
        print("Thank you for obeying the speed limit.")

main()
```

```
>>> import wk5l1_2
Please enter how fast you were going in mph: 34
Thank you for obeying the speed limit.
>>> wk5l1_2.main()
Please enter how fast you were going in mph: 55
You were going over the speed limit! Please slow down!
>>>
```



# else-if example

```
def main():
    speed = eval(input("Please enter how fast you were going in mph: "))

    if speed>35:
        print("You were going over the speed limit! Please slow down!")
    elif speed == 35:
        print("You were driving at the speed limit. Careful not to go too fast.")
    else:
        print("Thank you for obeying the speed limit.")

main()
```

# else-if example

```
def main():
    speed = eval(input("Please enter how fast you were going in mph: "))

    if speed>35:
        print("You were going over the speed limit! Please slow down!")
    elif speed == 35:
        print("You were driving at the speed limit. Careful not to go too fast.")
    else:
        print("Thank you for obeying the speed limit.")

main()
```

```
>>> import wk5l1_3
Please enter how fast you were going in mph: 35
You were driving at the speed limit. Careful not to go too fast.
>>> wk5l1_3.main()
Please enter how fast you were going in mph: 50
You were going over the speed limit! Please slow down!
>>> wk5l1_3.main()
Please enter how fast you were going in mph: 25
Thank you for obeying the speed limit.
>>>
```

---



# Think-pair-share

- Does it matter what order you put the elif conditions in? What effect, if any, does order have on process control?
- What are potential uses for these elif statements? Can you think of things we have written so far that would have been made easier by if statements?



# Combining conditions

- What if I want to know if multiple things are true at the same time?
- Python has syntax for AND as well as OR.
- Python allows multiple values in the condition

```
>>> x= -4
>>> x= 2
>>>
>>>
>>> x = -4
>>> y = 2
>>>
>>> if x < 0 and y > 0 :
>>>     print("Woohoo!")
```

```
Woohoo!
>>> if x < 0 or y > 0 :
>>>     print("Woohoo!")
```

```
Woohoo!
>>> if x < y < 3:
>>>     print("Woohoo!")
```

```
Woohoo!
>>>
```

# Using if statements for special cases

- Can use to protect against special cases
  - Example: user entered a value outside of the necessary input range
  - Example: program needs to take the square root of a number and can't do so if the number is negative
- Later, we will discuss a different way to do this: exception handling

## Example from homework

- Rewrite the grades program using if statements
- What is the minimum number of statements needed?
- Can you write the conditions in at least two different ways?
  - Describe the difference in how the computer goes through them

```
def main():
    #get filename from user and open file
    infilename = input("Enter the name of the file containing the scores: ")
    infile = open(infilename,"r")

    #open outputfile
    outfile = open("gradesif.txt","w")

    #read in lines and get scores
    for line in infile:
        firstname,lastname,score = line.split()
        score = eval(score) #turn score into a number from a string
        grade = ''
        if score >= 90 :
            grade = 'A'
        elif 90 > score >= 80:
            grade = 'B'
        elif 80 > score >= 70:
            grade = 'C'
        elif 70 > score >= 60:
            grade = 'D'
        else:
            grade = 'F'
        print(firstname,lastname,score,grade,file=outfile)

        print(firstname,lastname,"received a score of",score,"for a grade of",grade)

main()
```

## Conditions in this solution

- Most people write:

```
if score >=90:  
    grade = 'A'  
if 90 > score >= 80:  
    grade = 'B'
```

- Is this the simplest way to write this? Why or why not?





## Book question:

- Chapter 7 Question 1: Many companies pay time-and-a-half for any hours worked above 40 in a given week. Write a program to input the number of hours worked and the hourly rate and calculate the total wages for the week.
- Inputs? Outputs?
- Decisions? Can we write a formula?

```
1000
>>> def wages(hours,rate):
    pay = 0
    if hours>40 :
        pay = 40*rate+(hours-40)*rate*3/2
    else:
        pay = hours*rate
    return pay

>>> wages(40,10)
400
>>> wages(41,10)
415.0
```

## Book question:

- Textbook Chapter 7, Question 6: The speeding ticket fine policy for Podunksville is \$50 plus \$5 for each 5 mph over the limit, plus a penalty of \$200 for any speed over 90 mph. Write a program that accepts a speed limit and a clocked speed and either prints a message indicating the speed was legal or prints the amount of the fine, if the speed was illegal.
  - Can we write a mathematical function that would determine the fine?
  - Can we write it in terms of logical statements (if, if-else, if-elif)?
  - Can we implement it in a program?

# Book question: solution

```
>>> def speedtest(speedlimit,clockedspeed):  
    if(speedlimit>=clockedspeed):  
        print("The speed was within the legal speedlimit.")  
    else:  
        print("The speed was over the legal speedlimit.")  
        fine = 50 + 5*((clockedspeed-speedlimit)//5)  
        if clockedspeed > 90:  
            fine += 200  
        print("The fine will be $",fine)  
  
>>> speedtest(70,99)  
The speed was over the legal speedlimit.  
The fine will be $ 275  
>>> speedtest(25,21)  
The speed was within the legal speedlimit.  
>>>
```