Announcement

• Homework assignment #2 will be posted to the course website today

• Solutions to homework 1 are being distributed at the end of lecture today. ABSOLUTELY no late submissions are being accepted as of this point.
The most-recent “take home” exercise

Write python code that:

1. Prompts a user to input a number
2. Prompts a user to input a second number
3. Prompts the user to input the summation of the two numbers
4. Based on the user’s input and the “real” summation, outputs either of the two statements:

   'You are a math wiz' is a False Statement
   'You are a math wiz' is a True Statement

Task: Write a program that accomplishes the above task but uses if statement(s) as needed
Q: What does the code in the box output to the screen?

```python
print((3 == 5 or (3 != 5 and 5 != 7)) and 3 < 5)
```
Q: What does the code in the box output to the screen?

```
print((3 == 5 or (3 != 5 and 5 != 7)) and 3 < 5)
```

```
print((3 == 5 or (TRUE and TRUE)) and 3 < 5)
```

Q: When does the AND operator return true?
Q: What does the code in the box output to the screen?

```python
print((3 == 5 or (3 != 5 and 5 != 7)) and 3 < 5)
```

Q: When does the OR operator return true?
Q: What does the code in the box output to the screen?

```python
print((3 == 5 or (3 != 5 and 5 != 7)) and 3 < 5)
```

`TRUE`

Q: When does the AND operator return true?
From Last Time

Q: What does the code in the box output to the screen?

```python
print((3 == 5 or (3 != 5 and 5 != 7)) and 3 < 5)
```

TRUE

Q: When does the AND operator return true?
From Last Time

Q: What does the code in the box output to the screen?

```
print((3 == 5 or (3 != 5 and 5 != 7)) and 3 < 5)
```

```
print((3 == 5 or (TRUE and TRUE)) and 3 < 5)
```

```
print((3 == 5 or TRUE) and 3 < 5)
```

```
print(TRUE and TRUE)
```

```
print(True)
```
From Last Time

How many checks does the or operator perform?
How many checks does the and operator perform?
How many checks does the not operator perform?
From Last Time

How many checks does the or operator perform?
How many checks does the and operator perform?
How many checks does the not operator perform?

And: \( a \text{ and } b \)
Or: \( a \text{ or } b \)
Not: \( \text{not } b \)
From Last Time

How many checks does the or operator perform?
How many checks does the and operator perform?
How many checks does the not operator perform?

And: \( a \text{ and } b \)
Or: \( a \text{ or } b \)
Not: \( \text{not } b \)

The \text{and} operator determines if BOTH of its \textit{operands} are true, so at most TWO checks are performed. However, if the first check of the left operand (\( a \)) is false, then the \textit{and} operator doesn’t even check the right (\( b \)) operator.
How many checks does the or operator perform?
How many checks does the and operator perform?
How many checks does the not operator perform?

And: \( a \text{ and } b \)
Or: \( a \text{ or } b \)
Not: \( \text{not } b \)

The or operator determines if at least one of its operands are true, so at most two checks are performed. However, if the first check of the left operand (\( a \)) is true, then the or operator doesn’t even check the right (\( b \)) operand.
From Last Time

How many checks does the or operator perform?
How many checks does the and operator perform?
How many checks does the not operator perform?

And: \( a \text{ and } b \)
Or: \( a \text{ or } b \)
Not: \( \text{not } b \)

The \textbf{not} operator determines just if the single \textbf{operator} is true or false.
Warmup

How many strings can be provided as arguments to the `print` function?
Today

Conditionals
Conditional

We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```python
if (isCloudy and isRaining) :
    print(“wear coat and use umbrella”)

if (isCloudy or isRaining) :
    print(“bring coat and umbrella just in case”)
```
We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```python
if (isCloudy and isRaining) :
    print(“wear coat and use umbrella”)
```

```python
if (isCloudy or isRaining) :
    print(“bring coat and umbrella just in case”)
```
We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```python
if (isCloudy and isRaining) :
    print(“wear coat and use umbrella”)

if (isCloudy or isRaining) :
    print(“bring coat and umbrella just in case”)
```
We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear.

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```python
if (isCloudy and isRaining) :
    print("wear coat and use umbrella")

if (isCloudy or isRaining) :
    print("bring coat and umbrella just in case")
```

* The parentheses are not required, but are used often to make the code easier to read.
Conditional

We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

Python syntax

An expression that evaluates to True or False

if (isCloudy and isRaining) :
    print("wear coat and use umbrella")

if (isCloudy or isRaining) :
    print("bring coat and umbrella just in case")

It is syntactically incorrect to write if (m = 6) because the portion of code inside of the parentheses is an assignment, and does NOT evaluate to either True or False
We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear.

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```python
if (isCloudy and isRaining):
    print("wear coat and use umbrella")

if (isCloudy or isRaining):
    print("bring coat and umbrella just in case")
```

The colon indicates “done with the conditional” part of the code.
We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear.

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```python
if (isCloudy and isRaining) :
    print(“wear coat and use umbrella”)

if (isCloudy or isRaining) :
    print(“bring coat and umbrella just in case”)
```

Python syntax
The indented portion of code after the colon is called a code block.
We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear.

Q: If it is cloudy \textbf{and} rainy, print “wear coat and use umbrella”
Q: If it is cloudy \textbf{or} rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```python
if (isCloudy and isRaining) :
    print(“wear coat and use umbrella”)  

if (isCloudy or isRaining) :
    print(“bring coat and umbrella just in case”)  
```

Python syntax

The indented portion of code after the colon is called a **code block**.

Indent using either spaces or by tabbing.
Conditional

We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```python
if (isCloudy and isRaining) :  
    print(“wear coat and use umbrella”)

if (isCloudy or isRaining) :  
    print(“bring coat and umbrella just in case”)
```

Python syntax

The indented portion of code after the colon is called a code block

Indent using either spaces or by tabbing

Not indenting will generate an “expected an indented block” error
We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear.

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```
if (True):
    print("wear coat and use umbrella")
```

```
if (True):
    print("bring coat and umbrella just in case")
```
We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear.

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

Behavior of unary selection

- If the conditional evaluates to true, the code block is executed.
Conditional

We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```
Behavior of unary selection

If the conditional evaluates to false
```

```
if False:
    print("wear coat and use umbrella")

if False:
    print("bring coat and umbrella just in case")
```
We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

Behavior of unary selection

If the conditional evaluates to false

The code block is NOT executed
We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear

Q: If it is cloudy \textbf{and} rainy, print “wear coat and use umbrella”
Q: If it is cloudy \textbf{or} rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```python
isCloudy = True
isRaining = False
if (isCloudy and isRaining) :
    print(“wear coat and use umbrella”)

isCloudy = True
isRaining = True
if (isCloudy or isRaining) :
    print(“bring coat and umbrella just in case”)
```
We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear.

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```python
isCloudy = True
isRaining = False
if (isCloudy and isRaining):
    print("wear coat and use umbrella")

isCloudy = True
isRaining = True
if (isCloudy or isRaining):
    print("bring coat and umbrella just in case")
```
We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```python
isCloudy = True
isRaining = False
if isCloudy and isRaining:
    print("wear coat and use umbrella")

isCloudy = True
isRaining = True
if isCloudy or isRaining:
    print("bring coat and umbrella just in case")
```
We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear.

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```python
isCloudy = True
isRaining = False
if (isCloudy and isRaining):    
    print("wear coat and use umbrella")

isCloudy = True
isRaining = True
if (isCloudy or isRaining):    
    print("bring coat and umbrella just in case")
```
We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear.

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```python
isCloudy = True
isRaining = False
if (isCloudy and isRaining):
    print("wear coat and use umbrella")

isCloudy = True
isRaining = True
if (isCloudy or isRaining):
    print("bring coat and umbrella just in case")
```
We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```python
isCloudy = True
isRaining = False
if (isCloudy and isRaining) :
    print("wear coat and use umbrella")
```

```python
isCloudy = True
isRaining = True
if (isCloudy or isRaining) :
    print("bring coat and umbrella just in case")
```
We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear.

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```python
isCloudy = True
isRaining = False
if (isCloudy and isRaining) :
    print("wear coat and use umbrella")
```

```python
isCloudy = True
isRaining = True
if (isCloudy or isRaining) :
    print("bring coat and umbrella just in case")
```
We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```python
isCloudy = True
isRaining = False
if (isCloudy and isRaining) :
    print(“wear coat and use umbrella”)
```

```python
isCloudy = True
isRaining = True
if (isCloudy or isRaining) :
    print(“bring coat and umbrella just in case”)
```
We ended last time with the following goal: You want to write a program that recommends what clothing and accessory items to wear.

Q: If it is cloudy and rainy, print “wear coat and use umbrella”
Q: If it is cloudy or rainy, print “bring coat and umbrella just in case”

For which I showed you the following two pieces of code:

```python
isCloudy = True
isRaining = False
if (isCloudy and isRaining):
    print("wear coat and use umbrella")

isCloudy = True
isRaining = True
if (isCloudy or isRaining):
    print("bring coat and umbrella just in case")
```
The conditional execution of a block of code is often-times described using a flowchart.

```
if (Boolean expression) :
    statement
```

Flowchart:

- Start
- Condition
- Statement
- End
The conditional execution of a block of code is often-times described using a flowchart.

If the Boolean expression evaluates to True, the statement is executed.

If the Boolean expression evaluates to False, the statement is not executed.
The conditional execution of a block of code is often-times described using a flowchart.
The conditional execution of a block of code is often-times described using a flowchart

```python
if (Boolean expression) :
    statement
```

After which the if conditional is said to “complete”
The conditional execution of a block of code is often-times described using a flowchart:

If the Boolean expression evaluates to False, the block of code associated with the conditional statement is not executed.
The conditional execution of a block of code is often-times described using a flowchart:

```python
if (Boolean expression) :
    statement
```

Then the statement is NOT executed, and the if conditional ends.
Conditional

Q: How many statements can appear in the code block of the if selection conditional?
Conditional

Q: How many statements can appear in the code block of the if selection conditional?

A: At least one
Python also has (as do most other languages) a feature to perform binary selection.

If it is raining print “bring an umbrella” otherwise print “do not bring an umbrella”

Task: Write python code that uses one or more `if` statements to output the appropriate text based on the value of a Boolean variable `isItRaining`.

(On the board exercise)
Python also has (as do most other languages) a feature to perform binary selection

If it is raining print “bring an umbrella” otherwise print “do not bring an umbrella”

Q: For the code written on the board, how many “checks” of the variable `isItRaining` is/are performed?

Q: How might we speed up the execution of the program?
Conditional

Python also has (as do most other languages) a feature to perform binary selection.

If it is raining print “bring an umbrella” otherwise print “do not bring an umbrella”

Task: Write python code with a binary selection clause.

Syntax:

```python
if (Boolean expression) :
    statement_1
else :
    statement_2
```
Conditional

Python also has (as do most other languages) a feature to perform binary selection

If it is raining print “bring an umbrella” otherwise print “do not bring an umbrella”

Task: Write python code with a binary selection clause

Syntax:
```python
if (Boolean expression) :
    statement_1
else :
    statement_2
```

This portion behaves just like unary selection
Conditional

Python also has (as do most other languages) a feature to perform binary selection

The added “else” portion and its block is executed when the if condition evaluates to False

If it is raining print “bring an umbrella” otherwise print “do not bring an umbrella”

Task: Write python code with a binary selection clause

```python
if (Boolean expression) :
    statement_1
else :
    statement_2
```

CSCI 141
Computer Programming I
Python also has (as do most other languages) a feature to perform binary selection.

The added “else” portion and its block is executed when the if condition evaluates to False.

Q: How should we modify the flowchart of unary selection to capture the behavior of the binary selection?
Conditional

Python also has (as do most other languages) a feature to perform binary selection

syntax

if (Boolean expression) :
  statement_1
else :
  statement_2

The added “else” portion and its block is executed when the if condition evaluates to False

Flowchart for binary selection

start

condition

True

statement_1

end

False

Notice that statement_1 or statement_2 are executed, but NOT both
Q: What does the code in the box output to the screen?

```python
if (4 == 3):
    print (3 != 4)
else:
    print ("not equal")
```
Q: What does the code in the box output to the screen?

```python
if (4 == 3):
    print (3 != 4)
else:
    print ("not equal")
```

The conditional is checked. Does it evaluate to True or False?
Q: What does the code in the box output to the screen?

```python
if (4 == 3):
    print (3 != 4)
else:
    print ("not equal")
```

The conditional is checked. Does it evaluate to True or False?

The conditional portion of the if clause evaluates to false, therefore the block for the “else” portion is executed.
Q: What does the code in the box output to the screen?

```python
if (4 != 3):
    print (3 == 4)
else:
    print ("not equal")
```
Q: What does the code in the box output to the screen?

```python
if (4 != 3):
    print (3 == 4)
else:
    print ("not equal")
```

The conditional is checked. Does it evaluate to True or False?
Q: What does the code in the box output to the screen?

```python
if (4 != 3):
    print (3 == 4)
else:
    print ("not equal")
```

The conditional is checked. Does it evaluate to True or False?

The conditional portion of the if clause evaluates to true, therefore the block for the “if” portion is executed.

What does 3 == 4 evaluate to?
Q: What does the code in the box output to the screen?

```python
if (4 != 3):
    print (3 == 4)
else:
    print ("not equal")
```

The conditional is checked. Does it evaluate to True or False?

The conditional portion of the if clause evaluates to true, therefore the block for the “if” portion is executed

What does 3 == 4 evaluate to?

Therefore the output of this code is the text False
Using the conditional feature of a programming language, we can write programs that ‘reason’ about complex statements.

Goal: We want to check if two numbers, x and y, are the same, if $x > y$, or $y > x$? Write code that achieves that task.

(on the board exercise)
Using the conditional feature of a programming language, we can write programs that ‘reason’ about complex statements.

Goal: We want to check if two numbers, x and y, are the same, if \( x > y \), or \( y > x \)? Write code that achieves that task.

```python
if (x == y) :
    print("x is equal to y")

if (x < y) :
    print("x is less than y")

if (y < x) :
    print("y is less than x")
```

Although this works, how many checks are done, regardless of whether x is equal to, smaller, or larger than y?

Sample solution
Using the conditional feature of a programming language, we can write programs that ‘reason’ about complex statements.

Goal: We want to check if two numbers, x and y, are the same, if $x > y$, or $y > x$? Write code that achieves that task.

```python
if (x == y):
    print("x is equal to y")
if (x < y):
    print("x is less than y")
if (y < x):
    print ("y is less than x")
```

Although this works, how many checks are done, regardless of whether x is equal to, smaller, or larger than y?

There are three checks. Task: Write code that has just two checks (on the board exercise).
Using the conditional feature of a programming language, we can write programs that ‘reason’ about complex statements.

Goal: We want to check if two numbers, x and y, are the same, if \( x > y \), or \( y > x \)? Write code that achieves that task.

```python
if (x == y) :
    print(“x is equal to y”)
else :
    print(“not equal”)
if (y < x) :
    print ("y is less than x")
```

The code on the left performs only two checks, but does it generate the correct output for all possible combinations of x and y?

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>44</td>
<td>44</td>
</tr>
<tr>
<td>-7</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
</tr>
</tbody>
</table>
Using the conditional feature of a programming language, we can write programs that ‘reason’ about complex statements.

Goal: We want to check if two numbers, x and y, are the same, if x > y, or y > x? Write code that achieves that task.

```python
if (x == y):
    print("x is equal to y")
else:
    print("not equal")
if (y < x):
    print ("y is less than x")
```

Q: Is it possible to write code that in the best case scenario performs only a single check?

The code on the left performs only two checks, but does it generate the correct output for all possible combinations of x and y?

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>44</td>
<td>44</td>
</tr>
<tr>
<td>-7</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
</tr>
</tbody>
</table>
Using the conditional feature of a programming language, we can write programs that ‘reason’ about complex statements.

Goal: We want to check if two numbers, \( x \) and \( y \), are the same, if \( x > y \), or \( y > x \)? Write code that achieves that task.

```python
if (x == y):
    print("x is equal to y")
else:
    if (x < y):
        print("x is less than y")
    else:
        print ("y is less than x")
```

Sample solution

How many checks are performed?
Using the conditional feature of a programming language, we can write programs that ‘reason’ about complex statements.

Goal: We want to check if two numbers, x and y, are the same, if \( x > y \), or \( y > x \)? Write code that achieves that task.

Sample solution

```python
if (x == y):
    print("x is equal to y")
else:
    if (x < y):
        print("x is less than y")
    else:
        print("y is less than x")
```

Task: Draw the flow diagram for the code in the box (on the board exercise)
Write python code that:

1. Prompts a user to input how many cups of coffee have been consumed today
2. Prompts a user to input how many homework assignments are due tomorrow
3. If more than 10 cups of coffee have been consumed, then the program should output 
   
   **Enough caffeine to complete all assignments**

4. If at least 5 cups of coffee have been consumed, then the program should indicate that 1/2 (integer division) of the assignments due will be completed
5. If 4 or fewer cups of coffee have been consumed, then the program should indicate that 1/3 (integer division) of the assignments due will be completed

```
Cups of coffee drank today? 7
Number of assignments due tomorrow? 5
Enough caffeine to complete only 2 assignments

Cups of coffee drank today? 3
Number of assignments due tomorrow? 43
Enough caffeine to complete only 14 assignments
```
Finishing up selection