CSC 102
INTRO TO PROGRAMMING
WITH PYTHON

LECTURE 6
ITS DEJA VU ALL OVER AGAIN
MICHAEL GROSSBERG
USER IN/USER OUT

☐ USER OUTPUT: COMPUTER TELLS HUMAN SOMETHING

☐ PRINT "HELLO WORLD!"

☐ USER INPUT: HUMAN TELLS COMPUTER SOMETHING

☐ ANSWER = RAW_INPUT("WHATS YOUR FAVORATE COLOR")
A TIP FOR YOU

Program to compute a 15% tip on a bill

#Get a string from user
bill_amount = raw_input("Enter Bill Amount: ")
tip = float(bill_amount)*0.15
total_with_tip = tip + float(bill_amount)

print "The tip is $"+ str(tip)
print "The total with tip is $"+ str(total_with_tip)
IN ACTION

This runs it

```
Jerusalem:lec06_examples michael$ python tip_calc.py
Enter Bill Amount: 22.59
The tip is $3.3885
The total with tip is $25.9785
Jerusalem:lec06_examples michael$
```
IF ONLY ....

☐ HOW TO USE if STATEMENTS FROM RUR-PLE DAYS?

```python
if clear_in_front():
    move()
```

☐ PROBLEM: NO ROBOT, NO CLEAR IN FRONT

☐ WHAT CAN WE TEST?
We can check if a variable satisfies a condition.

```
x = 1.5
print x < 1
print x > 1
print 1 < x
print 2 < x
print x == 1.0
print x == 1.5
print x != 1.5
print (1 < x) and (x < 2)
```
The $20 Million Bug

In Spring 1993, in the Operating System development group at SunSoft, we had a "priority one" bug report come in describing a problem in the asynchronous I/O library. The bug was holding up the sale of $20 million worth of hardware to a customer who specifically needed the library functionality, so we were extremely motivated to find it. After some intensive debugging sessions, the problem was finally traced to a statement that read:

```x==2;
```

It was a typo for what was intended to be an assignment statement. The programmer 's finger had bounced on the "equals" key, accidentally pressing it twice instead of once. The statement as written compared x to 2, generated true or false, and discarded the result.
IF WORKS WITH CONDITIONALS

- IF EXECUTES IF TRUE, DOESN'T IF FALSE (NOT TRUE)

```python
the_conditional_is_true = True
if the_conditional_is_true:
    print "Hey! The conditional is True."
print "we're done."
the_conditional_is_true = False
if the_conditional_is_true:
    print "Hey! The conditional is True."
print "we're done."
```
DANGER DANGER

INVISIBLE CHARACTERS WILL KILL YOU

☐ BE WARE OF
  ☐ END OF LINES ^M
  ☐ QUOTATION MARKS “
  ☐ APOSTROPHES ’
  ☐ DASHES

COPY PASTE
GO INSANE

TYPE YOUR CODE IN A TEXT EDITOR

Wednesday, September 22, 2010
IF WITH ANY EXPRESSION

☐ IF WILL TEST ANY EXPRESSION

☐ IF FAILS IF VARIABLE IS ZERO OR “LIKE ZERO”

☐ OTHERWISE TRUE

```python
x_not_zero = 1
if x_not_zero:
    print "X is not zero"

x_not_zero = 0
if x_not_zero:
    print "X is not zero"

x_not_zero = ""
if x_not_zero:
    print "X is not zero"

x_not_zero = "I am 0"
if x_not_zero:
    print "No you are not"

x_not_zero = []
if x_not_zero:
    print "X is not zero"
```
DONT USE IF WITH STATEMENT

STATEMENTS NOT ALLOWED HERE

```python
if x = 5:
    print "Never gets here due to Epic Fail"
```

```bash
Jerusalem:lec06_examples michael$ python epic_fail.py
File "epic_fail.py", line 1
  if x=5:
     ^
SyntaxError: invalid syntax
```
PYTHON KEYWORDS

and  
as  
assert  
break  
class  
continue  
def  
del  
elif  
else  
except  
exec  
finally  
for  
from  
global  
if  
import  
in  
is  
lambda  
not  
or  
pass  
print  
raise  
return  
try  
while  
with  
yield  

THEY ALL JUST WORK
FEAR OF FROGS

x = 1.2
y = "frog"

if x and not (y == "frog"):
    print "x is not zero and y isn't frog"
elif y == "frog":
    print "Oh NO! y== 'frog' "
else:
    print "X must be zero"
WHILE WE ARE AT IT

SEPARATES WITH SPACE
DON'T NEED TO CONVERT TO STR

x = 0
while x < 10:
    print "Loop number", x
    x += 1  # same as x = x + 1

$ python loop10.py
Loop number 0
Loop number 1
Loop number 2
Loop number 3
Loop number 4
Loop number 5
Loop number 6
Loop number 7
Loop number 8
Loop number 9
HOW ABOUT A REAL PROBLEM?

- **FIND ROOTS FOR PARABOLA**

- **EQUATION** \( ax^2 + bx + c = 0 \),

- **DISCRIMINANT** \( \Delta = b^2 - 4ac \).

- **NO ROOTS IF** \( \Delta < 0 \)

- **1 ROOT IF** \( \Delta = 0 \)

- **ELSE 2 ROOTS** \( x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \).
FINDING PRIMES

☐ X IS A PRIME IF NO Y < X

☐ SO THAT Y|X AND

☐ Y! = X AND Y! = 1
```python
number = raw_input("What number should I check for primeness? ")
# initialize
is_number_prime = True

# loop parameters
start = 2; end = int(number); step = 1

# initialize loop
x = start

# test
while x < end:
    if end % x == 0:
        is_number_prime = False
        # increment
        x += step

# print result
if is_number_prime:
    print "The number", x, "is a prime number."
else:
    print "The number", x, "is not a prime number."
```

**WASH, LATHER, RINSE, REPEAT**

**INITIALIZE**

**TEST**

**INCREMENT**
THANKS