# Python Programming and Using the UMBC GL system

February 3, 2020

# Today

Questions?

Quick review of what we did last Wednesday

Finish that material and do some Python Programming

Talk about gl.umbc.edu and the programming you'll do

Go to lab tomorrow!!!

Homework 1 will be due Monday, Feb. 10 at 11:59:59 pm

# Last week

Integrated Development Environment (IDE) - what I'll use during lectures to write code

- Not a mandatory part of this class, but a good tool to know

Elements of a Python program

- Variables
- Literals
- CONSTANTS
- Expressions
- Statements

To finish up - the material below is repeated from last week’s slides

# “Sides” of the assignment operator

“Left hand side” and “right hand side”

- Left hand side is before - to the left of - the equals sign. This is where the value of the expression will be store
- Right hand side is after the equal sign. Evaluate everything to the right of the equal sign, and the store that value in the variable on the left

  num_candy = 4 * 12

  4 * 12 = num_candy  X not legal

# Operators

Special symbols that perform defined operations:

- Mathematical
- Comparison/Relational
- Assignment
- Boolean/Logical/Conditional

# Mathematical Operators

+ - * / // % **

+ - addition; works as you would expect

- Subtraction; works as you would expect

* multiplication

/ floating point division - results in a float number

// Integer division. Only valid if you have two integers; produces an integer.

% modulo

** exponentiation

Some examples:

5/3

5//3

5%3

# Comparison Operators

< less than

<= less than or equal to

> greater than

>= greater than or equal to

== equal to

!= not equal to

Be careful that you don't confuse = and ==

= is the assignment operator; it sets the value on the right to the variable on the left

== is a test to see if what's on the left is equal to what's on the right

# Assignment operators

=

+=

* =

-=

/=

Some examples

num_candy = num_candy + 1

num_candy += 1

num_candy *= 2

num_candy -= 2

num_candy /= 2

# Boolean operators

and

or

not

A boolean is either True or Fals

Boolean operators take Boolean values, combine them and yield a single Boolean value

9 > 8 and 5 < 9

num_candy != 0 or choice = ‘yes’

# Practice

Set the variable meal_bill to 30 dollars and 51 cents. Then calculate a 20 percent tip on meal_bill, and print out the total amount due

Calculate a GPA.  Assign values to the number of hours earned and total quality points. Calculate the GPA as number of quality points divided by number of hours earned. Print the GPA.

# Input and Output

Initially, we will read in all input from the keyboard and print all results to the screen.

- We'll cover reading from files and printing to files later

Input is done with the "input" statement; output is done with the "print" statement

print (3 + 4)

print (3, 4, 3+ 4)

print()

print ("the answer is", 3 + 4)

# Examples

```
a = 10

b = a * 5

c = “your result is:’

print(c, b)
```

```
a = 10

b = a

a = 3

print(b)
```

# Input from the keyboard

If you expect the user to input a meaningful value, you have to tell him or her what you want. The input statement lets you enter a string to be printed out as a prompt

user_num = input(“please enter your student number:”)

print(user_num)

When the input statement is executed by the interpreter, the program stops until the user has entered the required data

In python 3, input is always entered as a string. Even if it’s supposed to be an integer or a floating point number

If the user types 10, you get the string “10” NOT the integer 10.

If the user types 42.75, you get the string “42.75” NOT the floating point number 42.75

You can’t do math on a string!!!

# Changing the type of a value

If a variable's value is the wrong type, you can change its type by casting it to the type you want

To change a string to an integer, use int(the value)

```
age = int(input("enter your age in years as a whole number:"))
```

To change a string to a floating point number, use float()

```
gpa = float(input("enter your GPA to 3 decimal places: ")
```

To change a number to a string, use str()

```
student_ID = str(s_ID)
```

Can you convert the type of a variable; e.g. if gpa is a string, is this legal?

```
gpa = float(gpa)
```

# Using gl.umbc.edu to do your homework

Windows users:

- You will use a tool called putty to remotely log in to gl.umbc.edu
- Download putty
- https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html (going to www.putty.org will redirect you to this site)
- Then see https://www.youtube.com/watch?v=DH3of3tuBxA
-

Mac users:

- You can directly SSH into gl.umbc.edu from a terminal window
- See https://www.youtube.com/watch?v=UTU-CTH_xLw
-

# About gl.umbc.edu

It's a Linux system

It uses the bash shell for command-line input

- A "shell" is a command line interpreter
  - There are a number of them - c shell; sh; korn shell; bourne shell
  - bash - the "bourne-again shell" - is pretty much the standard for Linux & Mac these days

Emacs is the file editor provided

Python3 is the command used to execute python programs

When you're logged in to gl, everything is executing on that remote server. The only thing happening on your system is the input & output through your Terminal or PuTTy window

Watch (https://www.youtube.com/watch?v=r0otsJZ1ry0 ) before lab if you have a chance; it covers the basics

- It's less than 4 minutes long; you can watch it during lab if necessary