April 1 2015

CPE 101: Data Types:

Types

X = 5

X is a name of a variable, and we store things to that value

*Int- integer, whole number*

Y = 5.123

*Float –integer and decimal parts*

Z= 5.0

*Float*

Greeting = ‘hello class’

*string – sequence of characters*

***USE DOUBLE QUOTES IF STRING CONTAINS APOSTROPHE***

*Boolean: something that can be true or false*

Operators

+, -, \*, \*\*(power operator), /, % mod operator, // floor division (take float and round it to the lower number)

lets say we have integer + integer, then we will always have an integer

why is the % useful?

Lets say went on run that took 3324 seconds

55min 24 sec

float + float = float

float + int = float

*Operator precedence*

*()*

*\*\**

*negation*

*\* / // % in left to right order in problem*

***+ -***

**example: 3 + 6/4 – 2\*\*2 +8/2.0/(2+1)-25%4  
3 + 1 – 4 +1.333-1**

storage error occurs when ran in python. This happens because of the conversion between base 10 and binary

create files in vim ect.

*How do you make files work together?*

Import file\_name

File.name/x

*Classes*

Usually created to represent an object in the real world; they represent things

Point class

How might we represent a point?

Two floats, 1 for x one for y

**Point.py**

**c**lass **P**oint (lowercase class, upper Point)

class Point:

def \_\_init\_\_(self,x,y)

self.x=x

self.y=y

Attributes

**use\_point.py**

import point or from point import\*

pi=point(data).Point(class)

Object: a point running through the class with a temporary name called self

Strings that go over multiple lines need “”” text “””

*Time*

Reference timer.py and use\_timer.py

*Functions*

1. Built in functions
   1. Abs()
2. Imported from module
3. Write your own

Def print\_hello():

Print ‘Hello Class!’

Print ‘bye!’

Call print\_hello()

Print\_mult(word,num)

Print world\*num

Print\_mult(‘Paul’,10)

Def calc\_area(x1,x2,y1,y2):

Area=abs

Return area

def is\_hot(temp)

return temp > 90