Lecture 15

##By professor grimson

We are defining an abstract data type. Which are like the in built data types: ‘int’, ‘float’

Cartesian points as lists:

p1 = [1,2]

p2 = [2,pi/2] which is polar for [0,2]

We don’t know if the point is in Cartesian or in polar coordinates.

Class: template for creating instances of an object

instances are created are called as functions:

class Cartesian:

pass

cp = Cartesian() ##this is an instance

We can give some characteristics/attributes to the class.

cp.x = 15

cp.y = 16

this will correspond to the Cartesian point (15,16)

now if we print cp then it says that it is an instance of Cartesian

we can print and operate on cp.x and cp.y as if they were normal variables.

keyword ‘is’

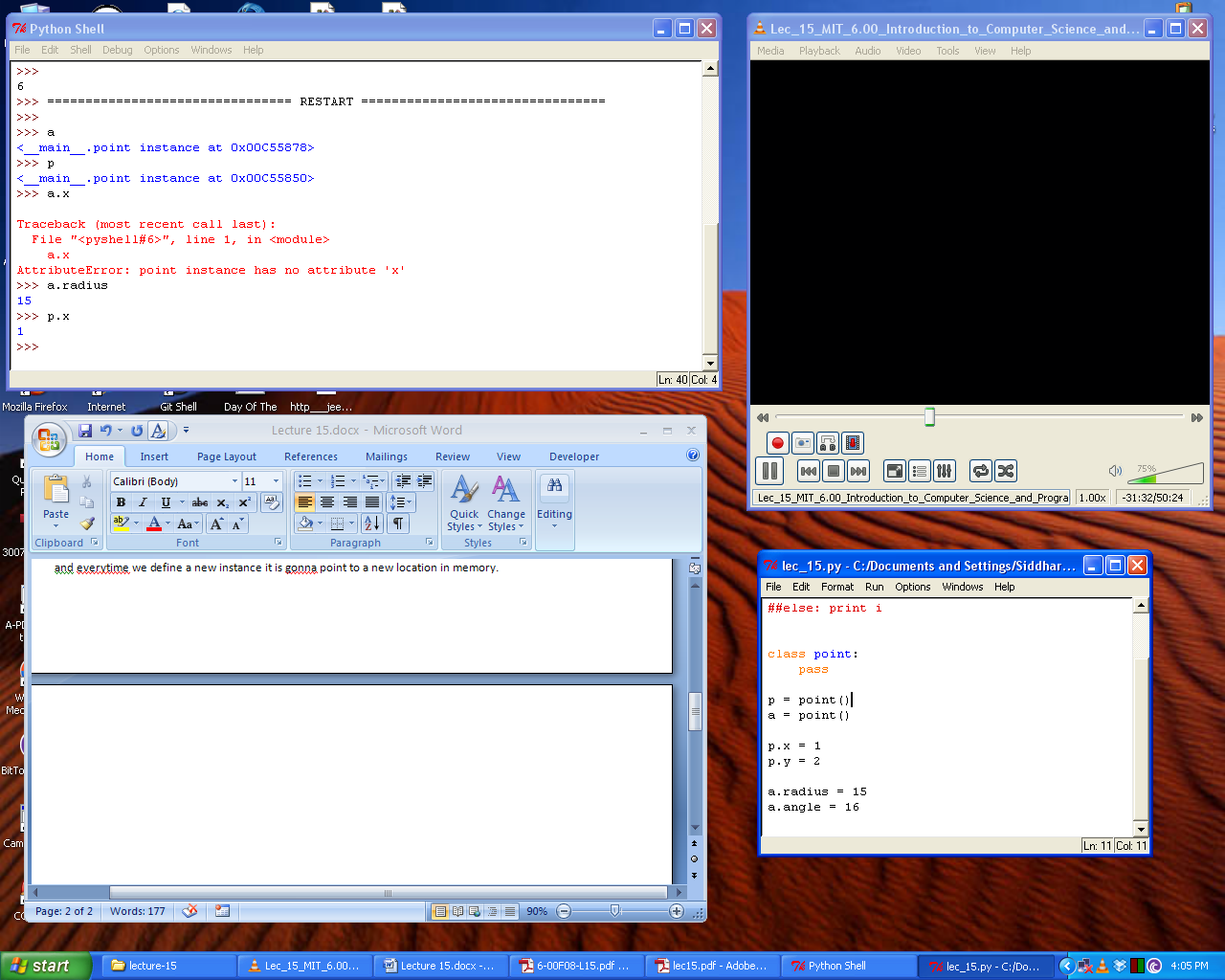
‘is’ does shallow/object equality.

a is b

will be true : if a and b point to the exact same location in memory.

whereas deep/value/set of values equality: we define it.

and everytime we define a new instance it is gonna point to a new location in memory.



here p has attributes x and y

and a has attributes has radius and angle attributes

when we call a class definition then the function \_\_init\_\_ is called.

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

when we call \_\_init\_\_:

* it is going to create an instance
* and it is going to use self to refer to that instance

24:00