PIC 16, Fall 2017

Q&A 4M

Monday, October 23, 2017

Matt Haberland

- In Python, is there an equivalent for "virtual" from C++?

No. In Python, *all* methods are virtual.

Another way to think of it is that there is no such thing as a superclass-typed reference/pointer to a subclass object. Because Python is dynamically typed, references are effectively typed the same as the object they refer to.

- What are the advantages/disadvantages of using:
  - `import moduleName`,
  - `from moduleName import *`, and
  - `python moduleName.py`

We talked about the differences a bit during the 10 a.m. lecture. Please review that video.

- Can a class have multiple parent classes to refer to? or can we refer to them in the methods without defining them as parent class?

Yes, please see the Q&A 4M Notebook for an example

- Can you explain about the use of "super"?

I can give examples. Please see the Q&A 4M Notebook

- How does encapsulation work with Python classes?

Please see:
http://stupidpythonideas.blogspot.com/2014/01/python-doesnt-have-encapsulation.html

It answers the question from many angles much better than I can.

- What is the difference between `__init__` and `__new__`?

The preparation says that __new__ will automatically run while `__init__` will not.

Inheritance is to use the superclass' variables and functions. So why do we still use `__init__` rather than `__new__`?

`__new__` seems to be a faster and more efficient method.

- Could you talk more about the constructor `__new__`? What are the contents typically defined in `__new__`?
- could you explain what is the difference between `__init__` and constructor?

Please see: http://spyhce.com/blog/understanding-new-and-init