WELCOME TO CS-521!!!

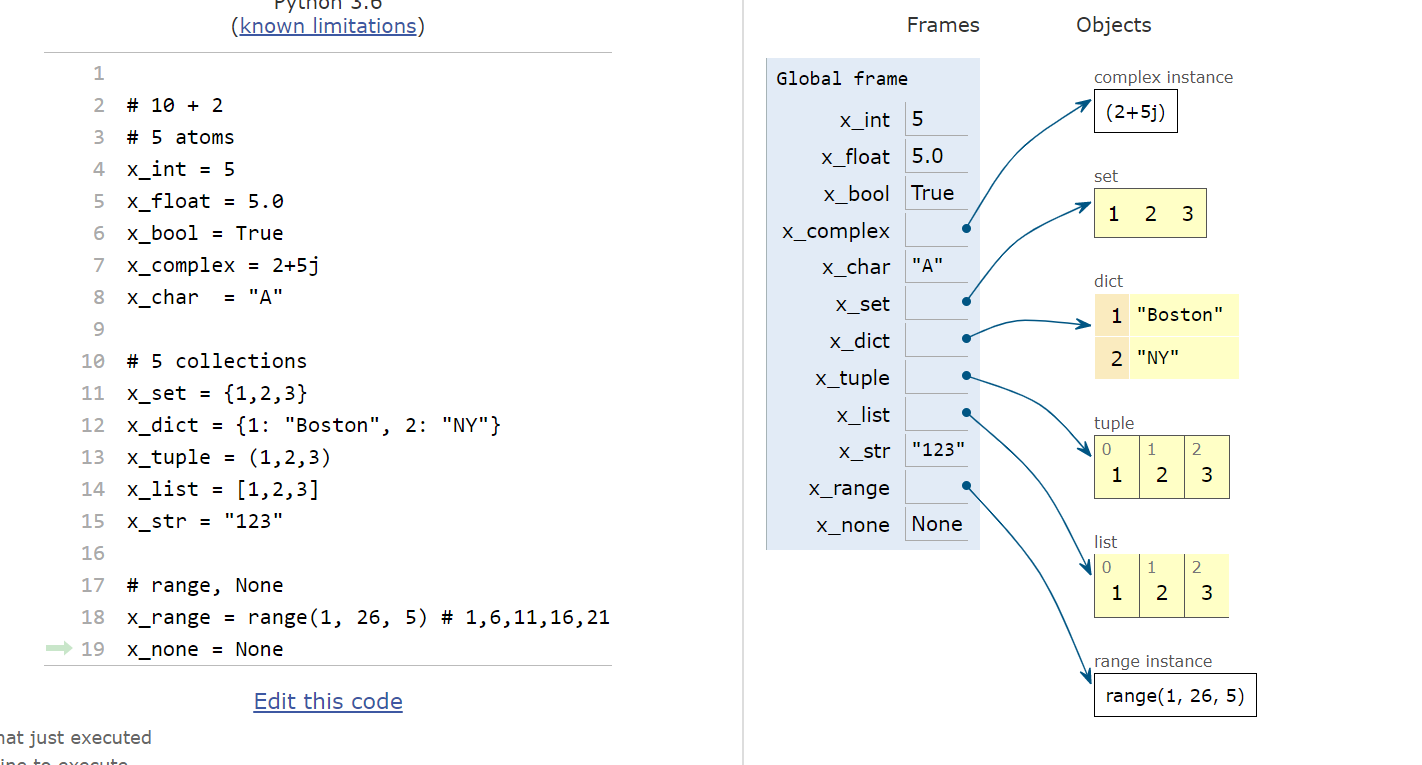
Tue - Thu, 6-7:30 p.m.

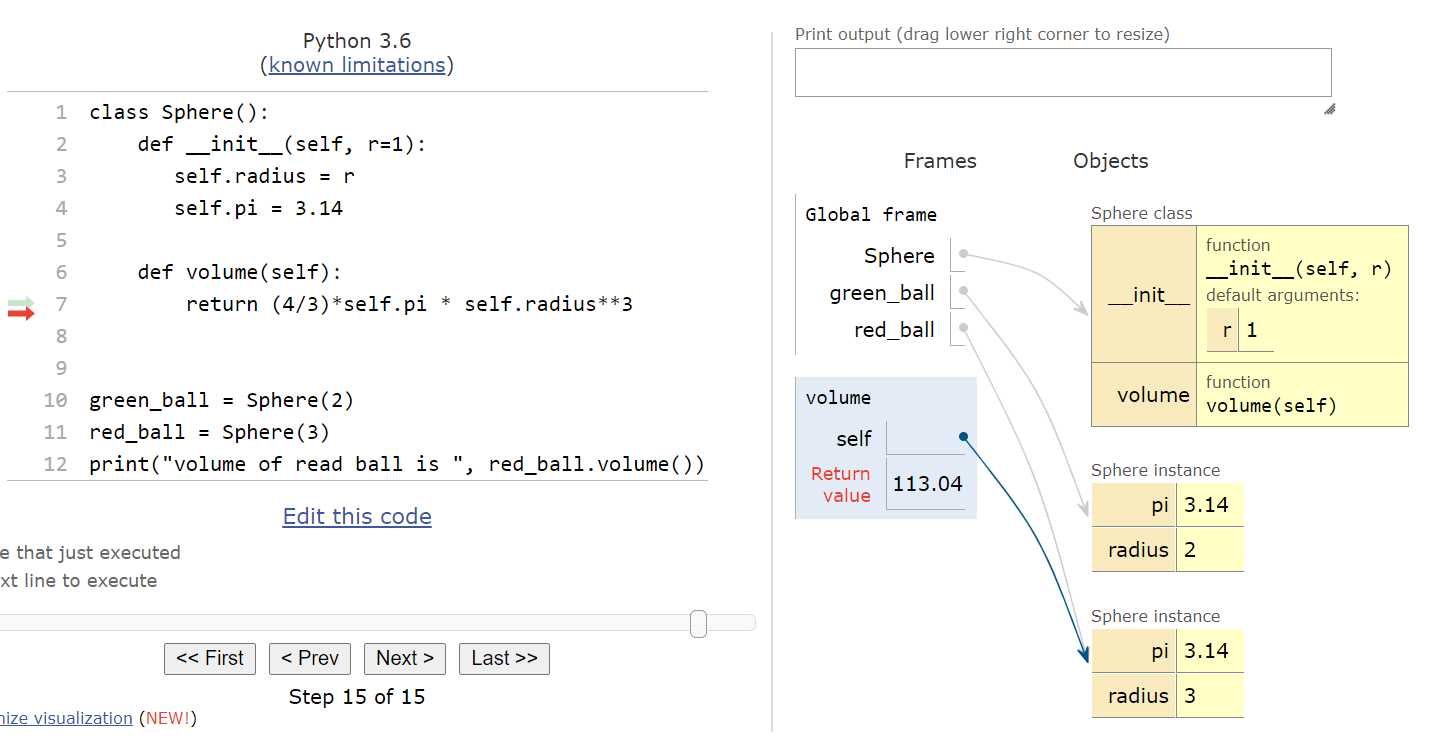
Lecture 12 (Dec 10, 2020)

Week 6:

Homework 6

Quiz 6





in the above definition, we have multiple copies of the same value “self.pi”

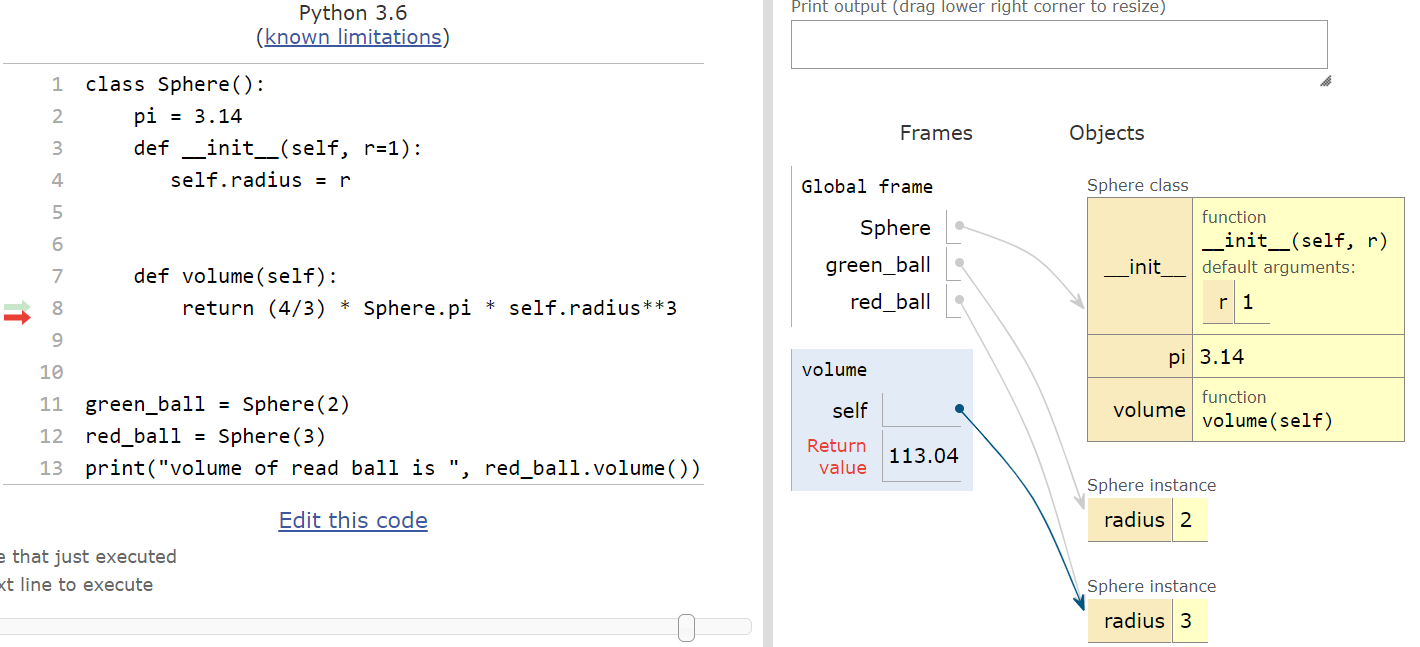
This is inefficient

want: single copy of pi shared across instances

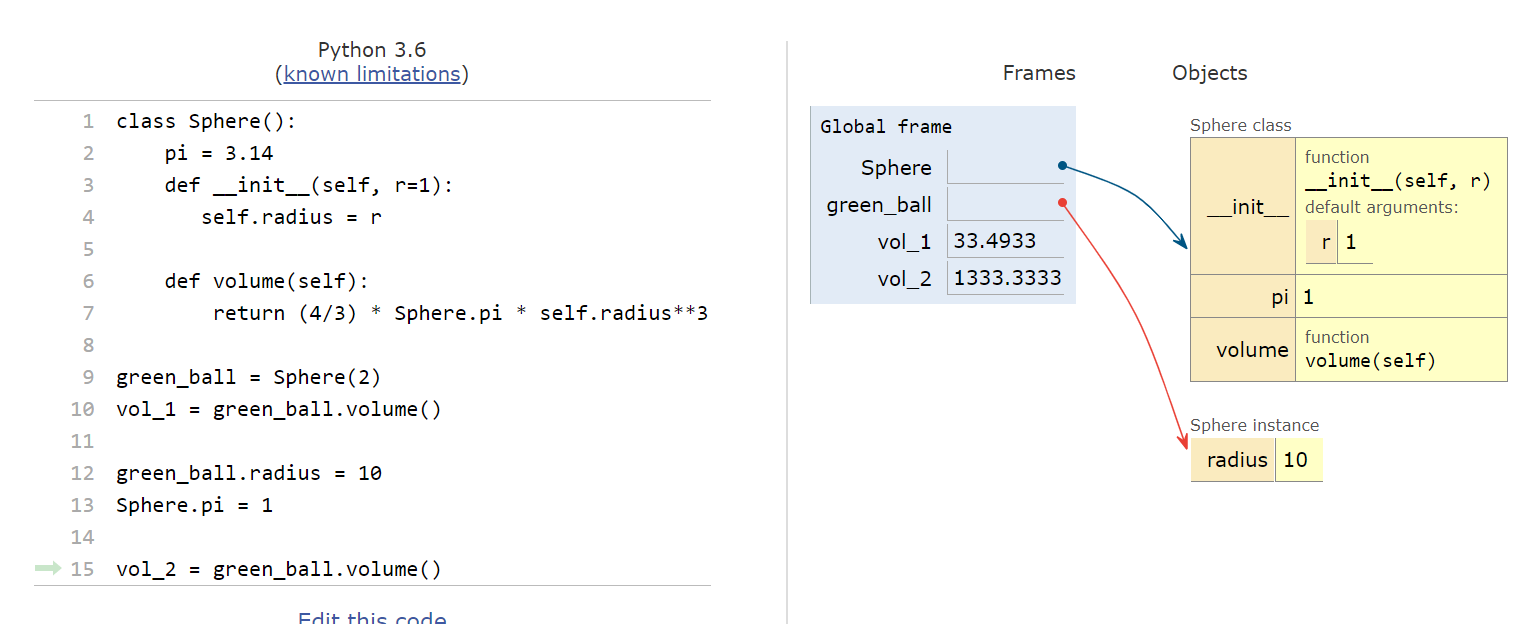
we want to make pi “static”

in Python we define variables after “class” keyword and before \_\_init\_\_()

this makes such variables “static”

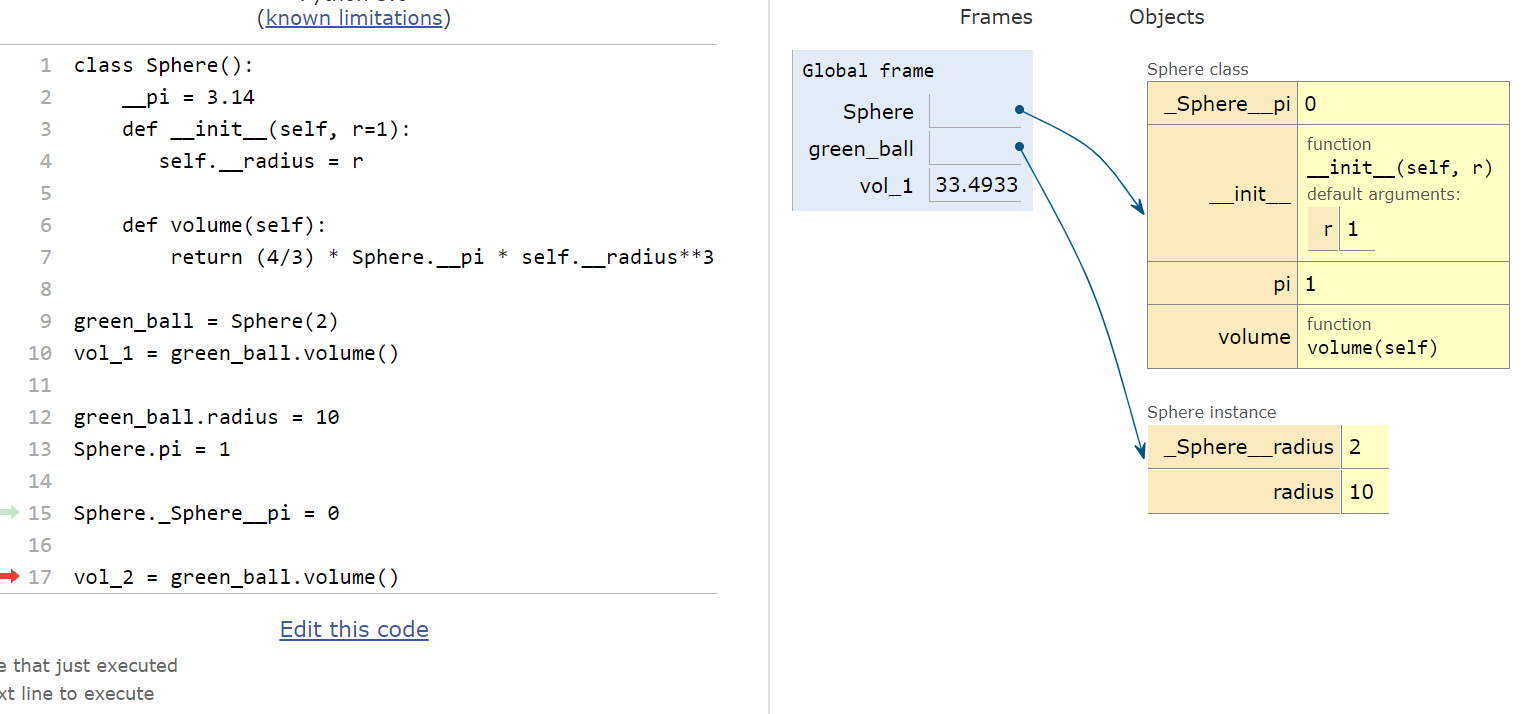


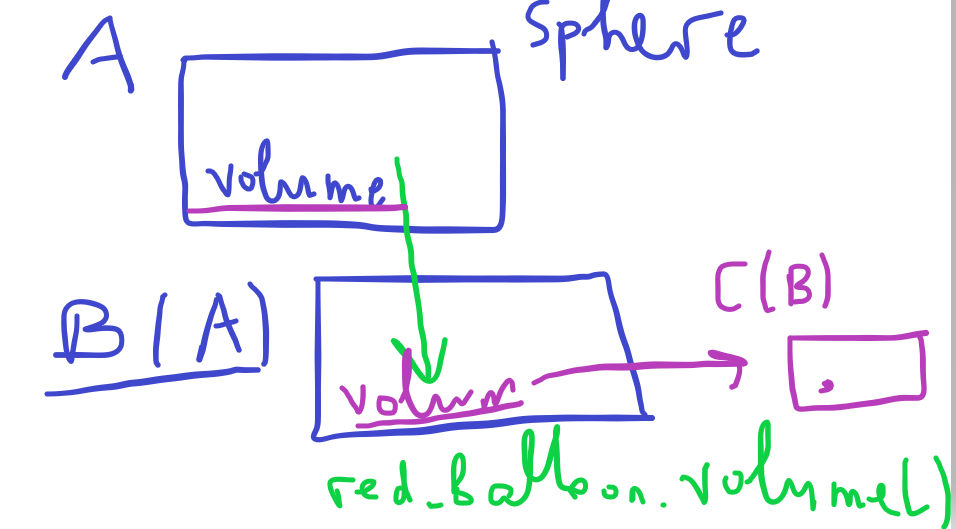
in python, there is a limited mechanism to prevent a user from accessing variables directly



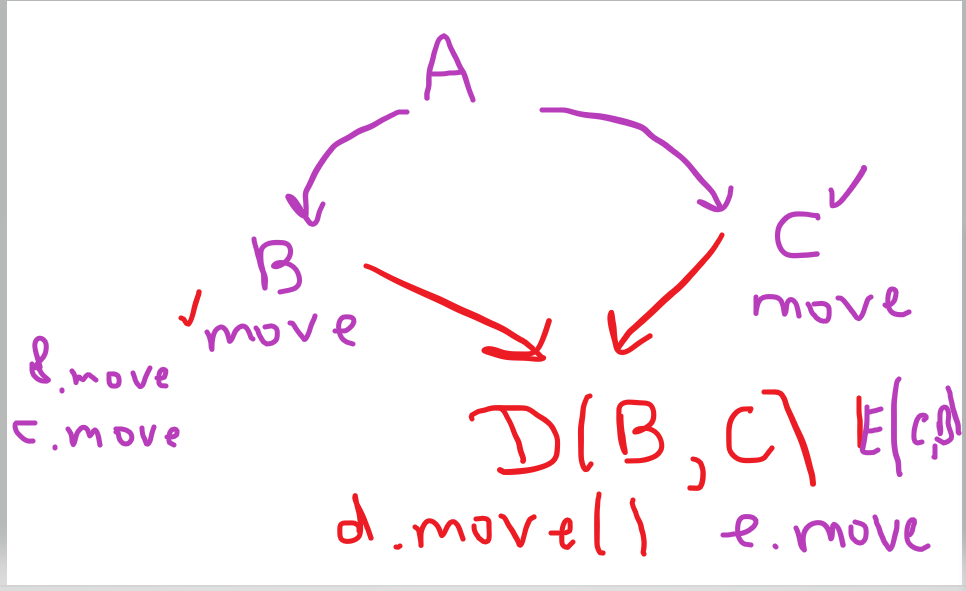
unlike C++ or Java we do not have private or public variables

Python has a limited mechanism: name mangling





you can call a method on an instance of class C. If that method is not defined in C, then Python will look for that method in B (parent of C). If it is not defined there, it will look in the parent of B and so on.



Python allows multiple inheritance

this creates the following problem: if you have a method with the same name defined in both parents (in our example, “move”),

and you have an instance of class “D”, what does d.move() mean?

Answer: the order in which the parents are specified is important

EXAMPLE:

we will extend Python list as follows:

1. create a sub-class derived from “list”
2. call this Counter\_List
3. just like a Python list but we will maintain a counter as to how many times indexing is used on such a list

