

New York University
School of Continuing and Professional Studies
Division of Programs in Information Technology

Introduction to Python
Homework Discussion, Session 10

10.1 Create a class, Square, each of whose instances (i.e., each Square object) have an attribute which is an integer. Each Square object has the following methods:

- a. `__init__()`: set the integer attribute to 2.
- b. `def squareme(self)`: square the integer attribute and store the new squared value in the same attribute, replacing the old value (i.e. the first time you call it, it will replace 2 with 4).
- c. `def getme(self)`: return the integer attribute.

The exercises preceding this assignment take you through three important components of this solution:

- the creation of an instance
- the use of a method to set an instance attribute (i.e. `self.attr = value`) and a method to return the value of the attribute
- the use of `__init__` to initialize an attribute in the instance

The solution for this homework will be very similar to those exercises. `__init__()` will need to set the instances attribute to 2 (so the value can be properly squared multiple times). `squareme()` will simply square the value of the attribute and assign it back to the attribute; `getme()` will simply return the attribute.

EXTRA CREDIT

10.2 List of a Maximum Size. Create a class, `MaxSizeList`, that constructs instances that have a list as an attribute (set as an empty list in `__init__`). The constructor takes an integer as an argument, which becomes the maximum size of the list (stored as a second attribute).

Since the instance/object must be aware of its max size and it must also maintain a list of values, it will have two attributes: an integer and a list (i.e. `self.maxsize`, `self.thislist`). The list will be initialized in `__init__` as an empty list and the max size will be initialized in `__init__` as the value passed to the constructor.

Then in `push()`, we must determine whether the list size will exceed the max size if we append another value. Since both the max size and the list are stored in the instance, we can simply compare the length of the list to the max size value to see if a value needs to be removed from the list. To remove the first element of a list, use `list.pop(0)`, so in this case `instance.thislist.pop(0)` (where `thislist` is the name of the attribute used to store the list) before or after appending the new object.

`get_list()` simply returns the list stored in the instance.
