

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

Introduction to Python
Homework, 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 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.

Sample usage (your code must be able to behave as shown below, including being able to create multiple arguments with their own data -- PLEASE MAKE SURE YOUR OBJECTS HOLD THEIR OWN DATA -- MAKE SURE THEY CAN BEHAVE EXACTLY AS SHOWN BELOW, ESPECIALLY THE FINAL LINES OF THE EXAMPLE USAGE).

Also, make sure you can create any number of objects, not just three!

```
# all output must match!

n1 = Square()
n2 = Square()
n3 = Square()

n1.squareme()
val1 = n1.getme()
print(val1)           # 4

n2.squareme()
n2.squareme()
val2 = n2.getme()
print(val2)           # 16
print(n1.getme())     # 4

n3.squareme()
n3.squareme()
n3.squareme()
n3.squareme()
n3.squareme()
val3 = n3.getme()
print(val3)           # 4294967296

print(n1.getme())     # 4
print(n2.getme())     # 16
print(n3.getme())     # 4294967296
```

Expected Output:

```
4
16
4
4294967296
4
16
4294967296
```

SUPPLEMENTARY (EXTRA CREDIT) EXERCISE (try to do this if you can)

- 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).

Add a method, `push()`, which appends a value to the list. However, if the list has already reached its maximum size (as set in the constructor), remove the first element from the list before appending the new element to the list. (You can use the list method `pop()` with an argument of 0 to remove the first element of the list, or you can use a slice (`alist = alist[1:]`).

Sample code to use the class and expected output:

```
a = MaxSizeList(3)
b = MaxSizeList(1)

a.push("hey")
a.push("ho")
a.push("let's")
a.push("go")

b.push("hey")
b.push("ho")
b.push("let's")
b.push("go")

print(a.get_list())    # ['ho', "let's", 'go']
print(b.get_list())    # ['go']
```