CSE 30 4-13-21

- PA2 : friday
- lab1 : ext to tonight

continue .. Generators

Recursive fcn : GCD

Ex find gcd of 80, 45

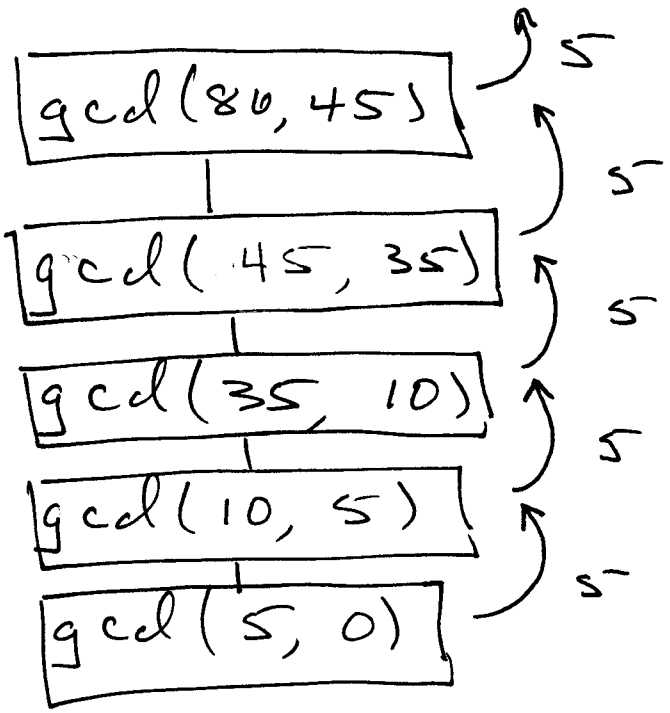
Euclids Algorithm:

a		b		r	
80	=	1 · 45	+	35	
45	=	1 · 35	+	10	
35	=	3 · 10	+	5	← gcd
10	=	2 · 5	+	0	← stop

↑ return b

```
def gcd(a, b):
    r = a % b
    while r > 0:
        a = b
        b = r
        r = a % b
    # end
    return b
# end
```

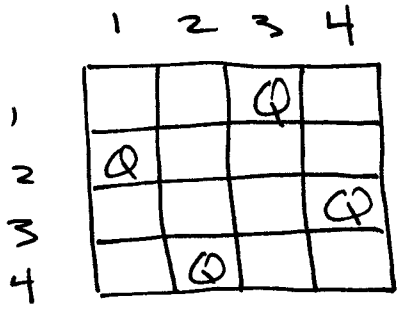
Recursive version: Box trace



Paz stuff:

n - Queens Problem

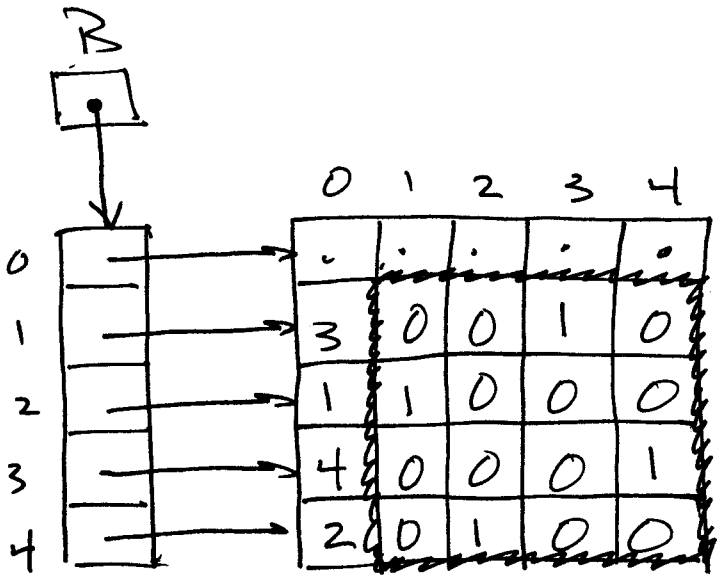
n = 4



a solution to 4-Queens

How to represent chessboard. ?

list of lists:



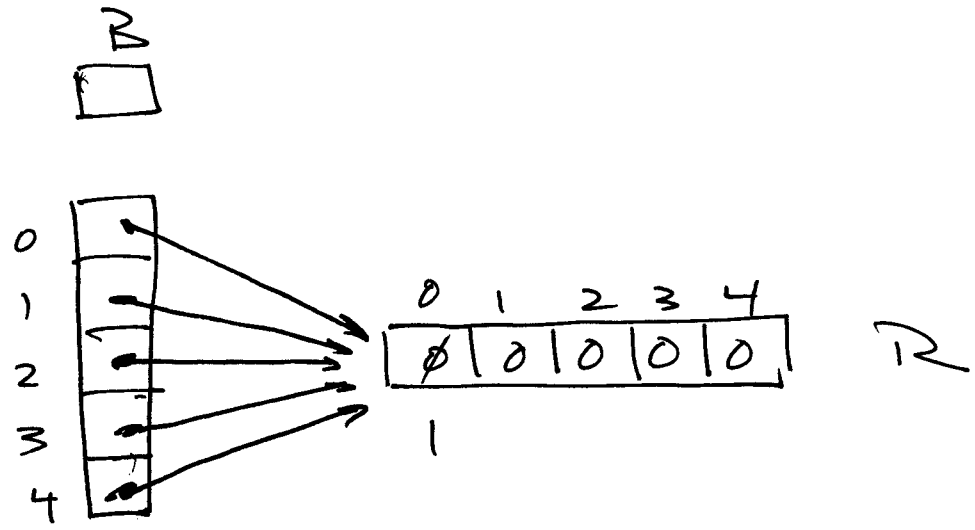
How to create B?

$B = 5 * [\,5 * [0]\,]$

$= 5 * [\,[0,0,0,0,0]\,]$ ← write a loop instead

Try $B[0][0] = 1$

we got this



Do something like

```
for ---------- :
    B.append( R[:] )
#end.
```

↑ copy of R

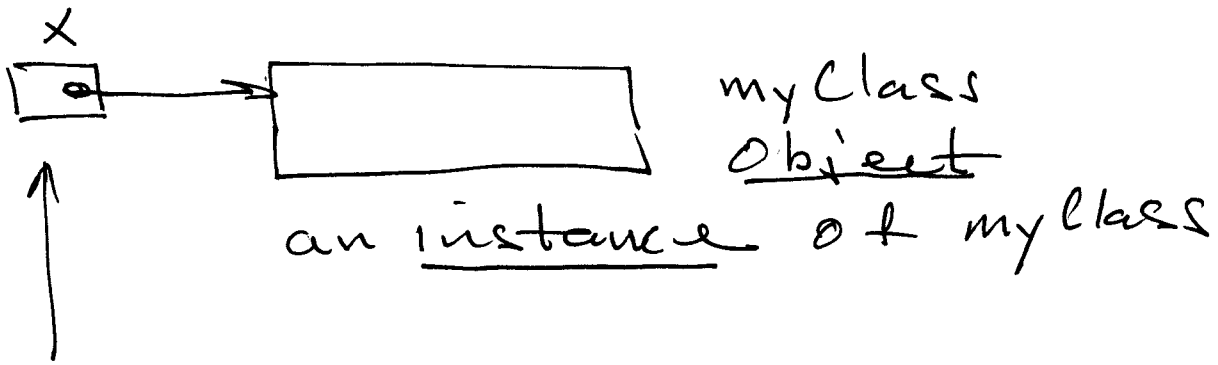
classes

classes are used to define new data types.

simplest class

```
class myClass:
   Pass
# end
```

```
x = myClass( ) # myClass constructor
```

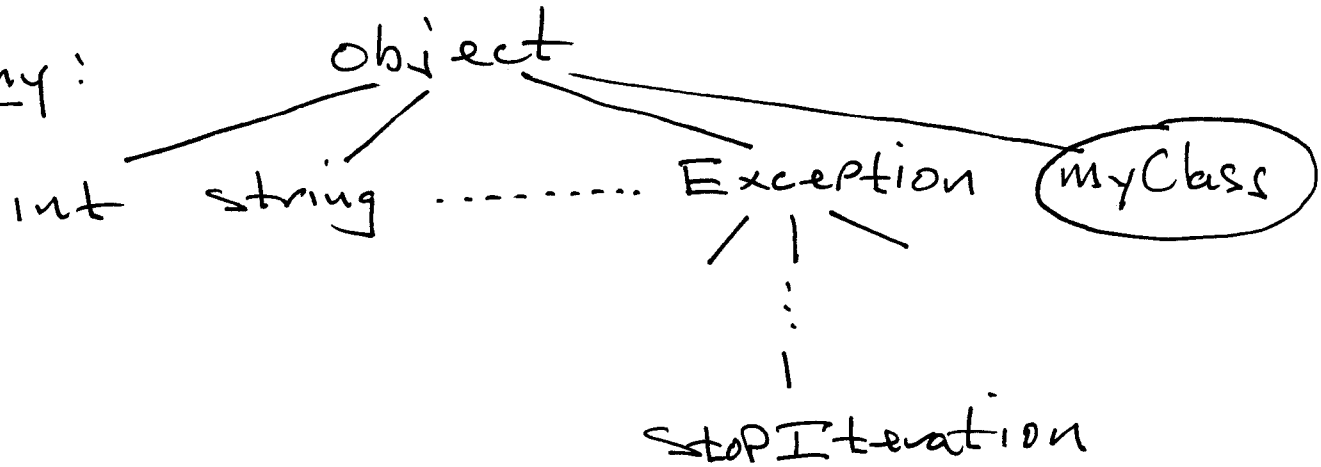


The members of a class are called its attributes. they can be

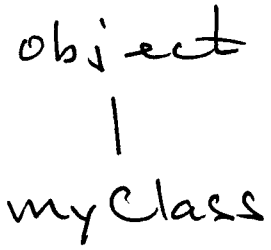
- Variables (properties): references to other objects
- methods (functions): operations on variables.

dir(x) or dir(myClass) we see a bunch of attributes inherited from object superclass:

hierarchy:



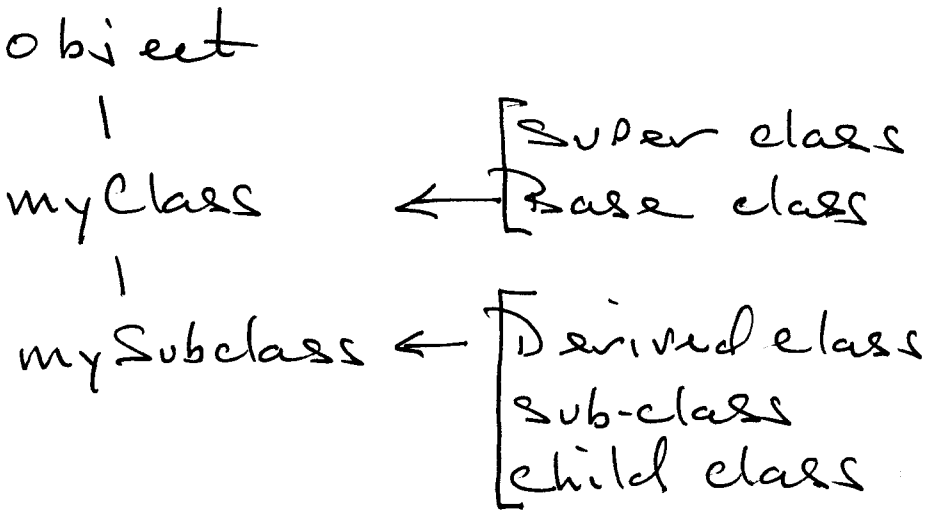
locally:



To create further subclasses do:

```
class mySubclass(myClass):
    Pass
#end
```

now:



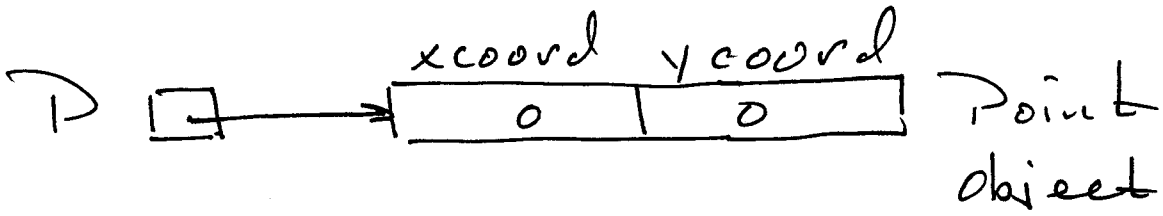
Inheritance: every attribute of Base class is also an attribute of Derived class.

Ex.

optional

```
class Point(object):
    xcoord = 0
    ycoord = 0
#end

P = Point()  # Point constructor
```




```
print(P.xcoord) # 0
  ..  (P.ycoord) # 0

P.xcoord = 1
P.ycoord = 7

print(P.xcoord) # 1
  ..  (P.ycoodl) # 7
```

Apparently an instance of a class is like a dictionary.