

cse 30 5-27-21

1

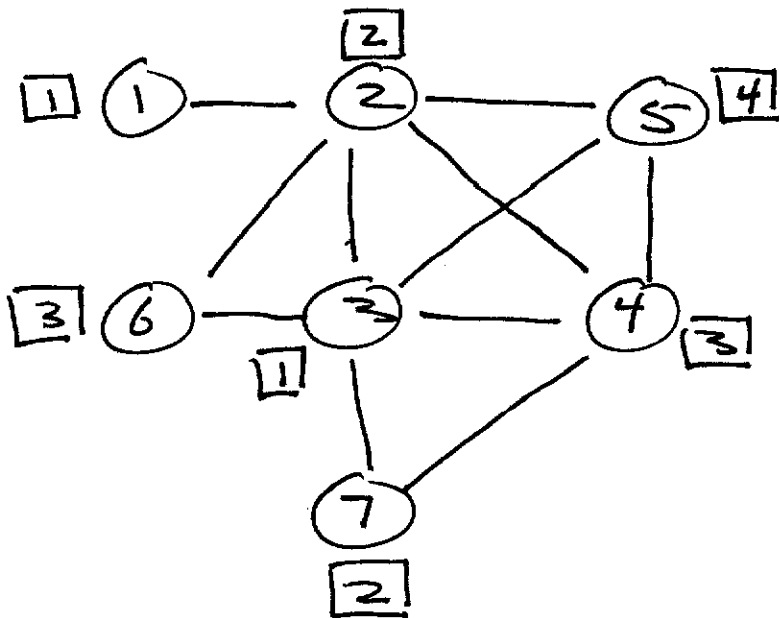
Pa6: ext. $\approx$ days $\rightarrow$ Tuesday.

Pa7: canceled.

lab4, 5: Probably not.

Pa6: Another coloring example

Ex $x: \bar{3} \quad \bar{2} \quad \bar{4} \quad \bar{5} \quad \bar{6} \quad \bar{7} \quad \bar{1}$
 $\text{deg: } 5 \quad 5 \quad 4 \quad 3 \quad 2 \quad 2 \quad 1$



$$\chi(G) = 4$$

colors used = $\{1, 2, 3, 4\}$

Note: Random Input generators for Pa6

- RandomInput1.py

input: n, m

output: random input file

- RandomInput2.py

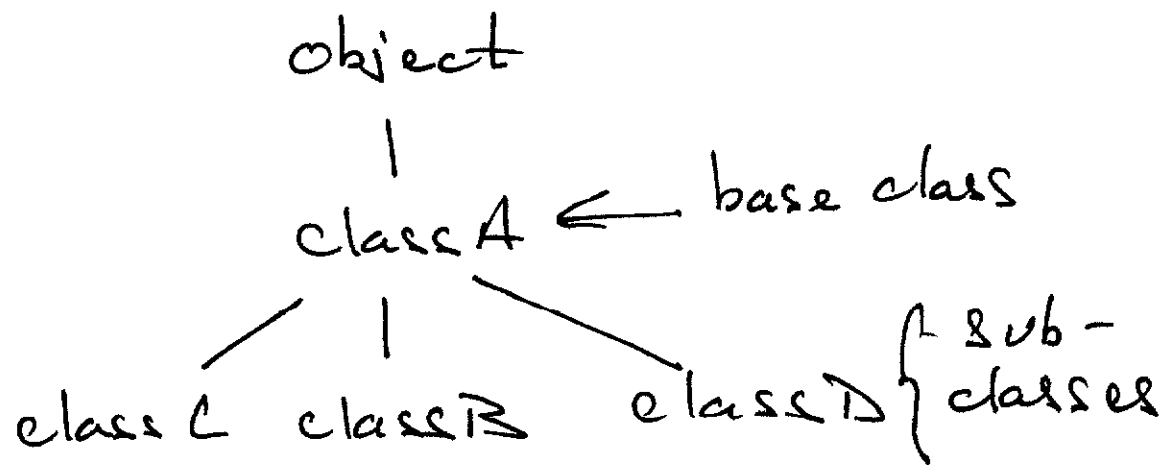
input: $n, X(G)$

output: matched pair random files

input file: n, X

output file: exact answer
to optimisation problem

Subclasses and Inheritance



- a subclass inherits all members of its base class
 - attributes (member variables)
 - methods (member functions)
- a subclass has the option to re-define its inherited methods : method overriding

Note:

• '...{ }...{ }...'.format(a, b)

is same as

{ }...{a}...{b}...

• $x = 6$

`type(x)` $\rightarrow$ `<class 'int'>`

`type(type(x))` $\Rightarrow$ `<class 'type'>`

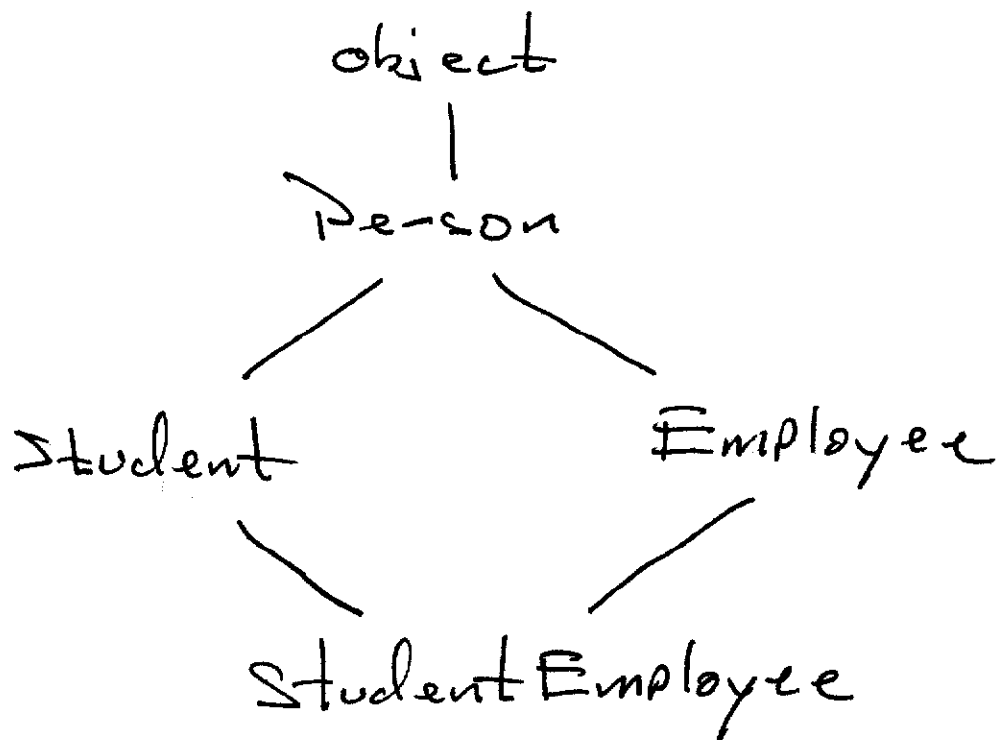
There is a class called 'type'
and `type()` is its constructor

`type(x).__name__` $\rightarrow$ 'int'

the class 'type' has an

attribute called `__name__`

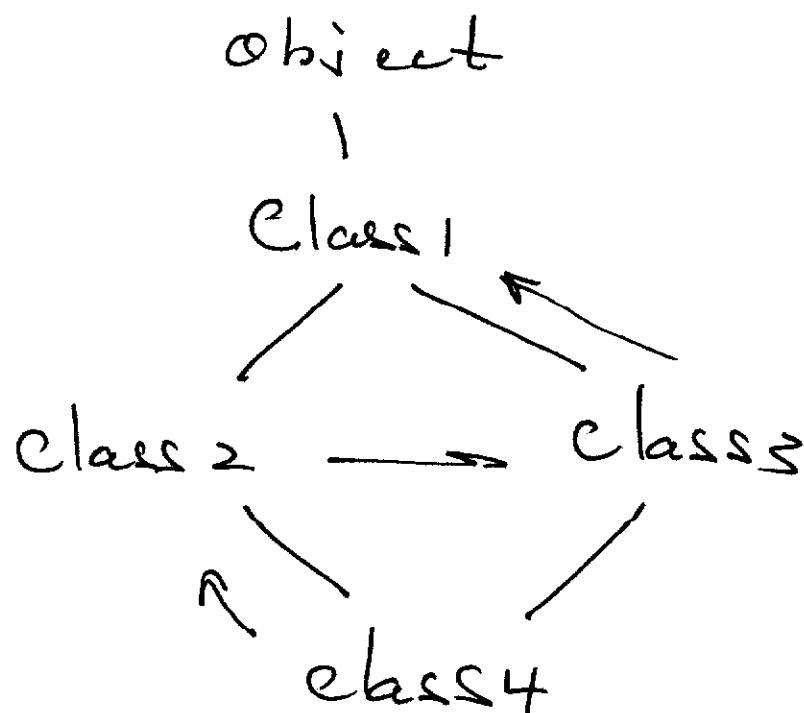
Ex. Person.py



Problem: If Student and Employee both override same Person method, in different ways, which one does StudentEmployee get.

- built-in function `super()` gets the super-class of an object.

Ex Multiple Inheritance. Py



Method resolution order: (for class 4)

class 4, class 2, class 3, class 1