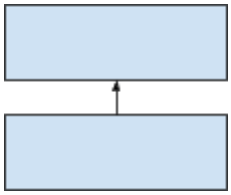
→ if given int, float or any other data, one was cast to ‘str’
→ del (destructor) method
   a. Gets activated just before process is sent for Garbage collection
   b. Usually coded to handle cleanup tasks like closing open connections, files, databases.
   c.
```
def __def__(self):
        # body
```
   d. Del method

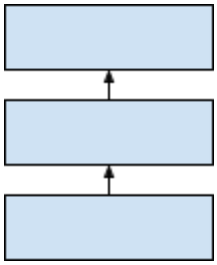
→ security / visibility / access specifier levels:
   a. Public
      i. Var
      ii. by default public
      iii. Can be accessed by anyone/anywhere
   b. Private
      i. _var
      ii. starts with single underscore
      iii. Not reflected, but if known accessed
   c. Strong private
      i. __var
      ii. starts with double underscore
      iii. Not reflected, not accessed outside
      iv. Only accessed within class by member only
   d. as
→ abstraction: never give out data, but give data to methods only
→ objects are always passed as reference
→ membership
   a. Local membership: variable exists only within method, and gets removed when method is over
   b. Instance membership/object member:
      i. anything with ‘self.’ e.g. self.name
      ii. Every object has its own call
      iii. They exist till object exists
   c. Class member:
      i. store and access with class
      ii. Object can access, but cannot store
      iii. Exists till class is loaded in memory
   d. as
→ first come first serve is followed in inheritance in python
→ we use bracket inherit super class to sub class
→ public & private can be inherited, but strong private cannot be inherited

→ Types of inheritance

a. Single level

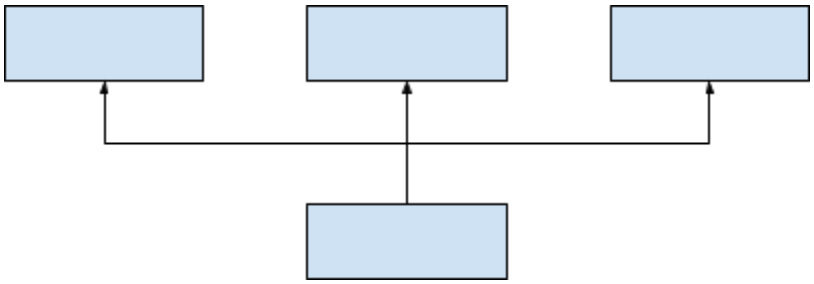


b. Multi-level

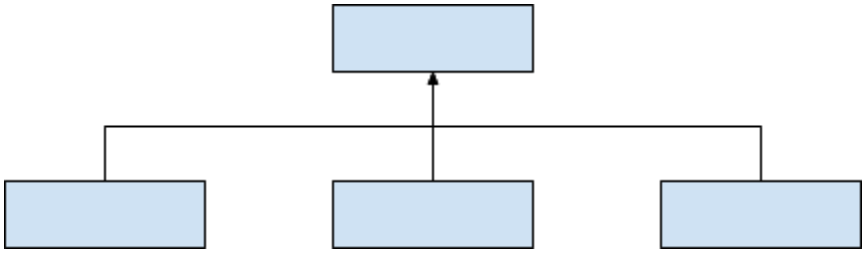


c. Multiple

i. First-come first-serve is followed to avoid ambiguity



d. Hierarchical



→ overriding:

a. re-writing super-class' methods in subclass, due to which it'll reject inheritance

→ pass:

a. 'pass' in python is a no-operation statement, used as pillar
b. Generally replaced by code later-on

→ setattr() & getattr() can create properties within NULL class object

→ getattr() can even get private data as well as strong private data
→ setattr() and getattr() can also work with other options created by pre-existing classes