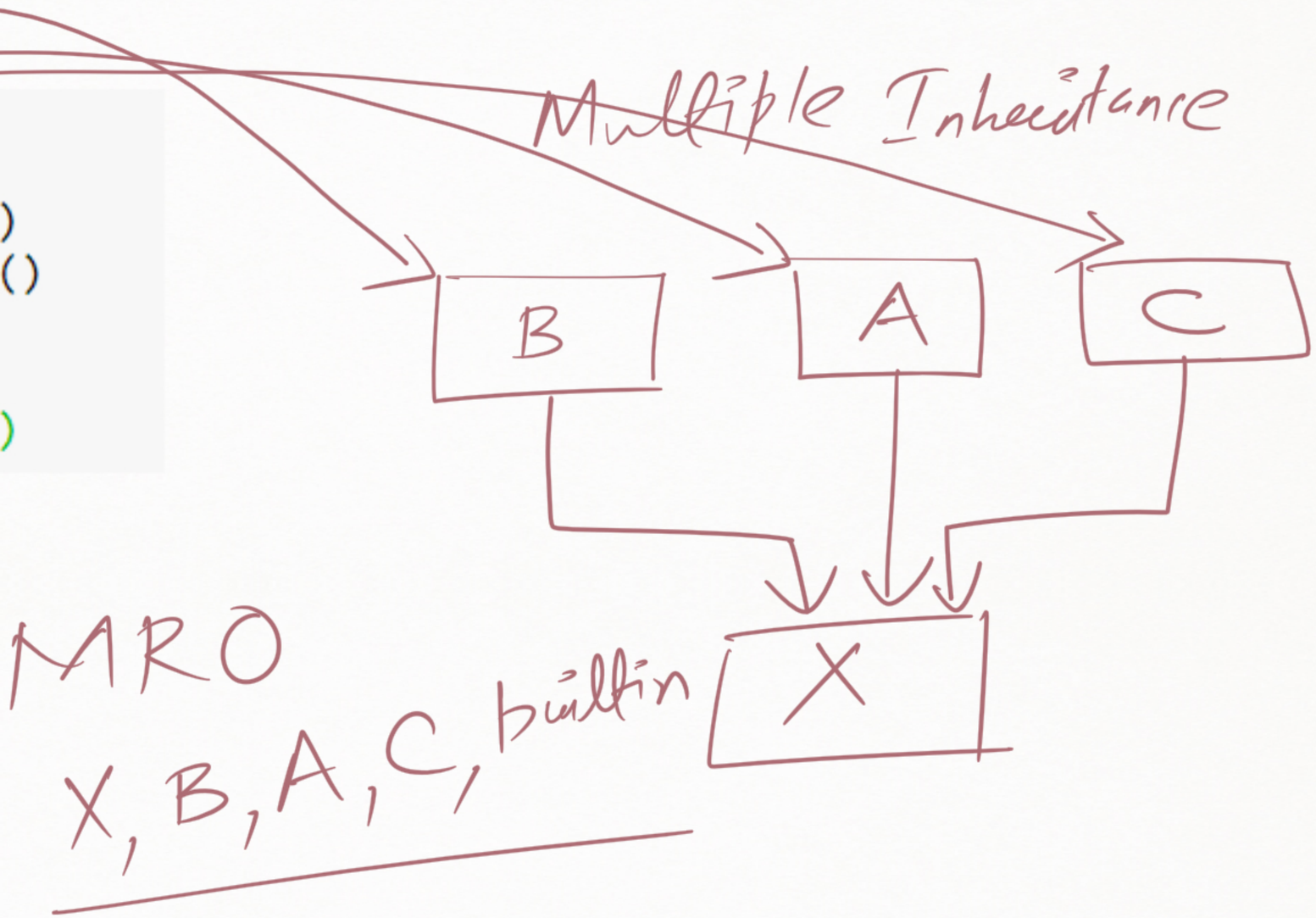
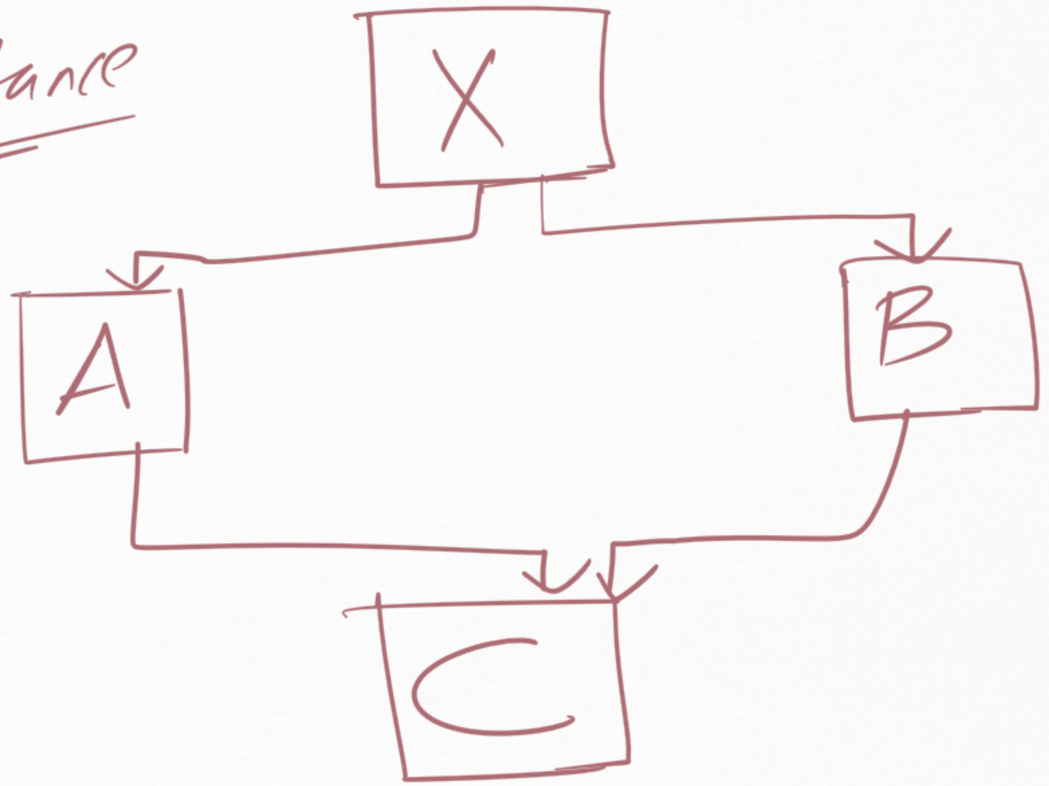
```python
class X(B,A,C):
    def __init__(self):
        print("inside X")
        super().__init__()

    def fun(self):
        print("fun of X")
```



```
:  class X:
       def __init__(self):
           print("inside X")

       def fun(self):
           print("fun of X")

   class A(X):
       def __init__(self):
           print("inside A")

       def fun(self):
           print("fun of A")

   class B(X):
       def __init__(self):
           print("inside B")

       def fun(self):
           print("fun of B")

   class C(A,B):
       def __init__(self):
           print("inside C")

       def fun(self):
           print("fun of C")
```

Hybrid Inheritance



MRO = C, A, B, X

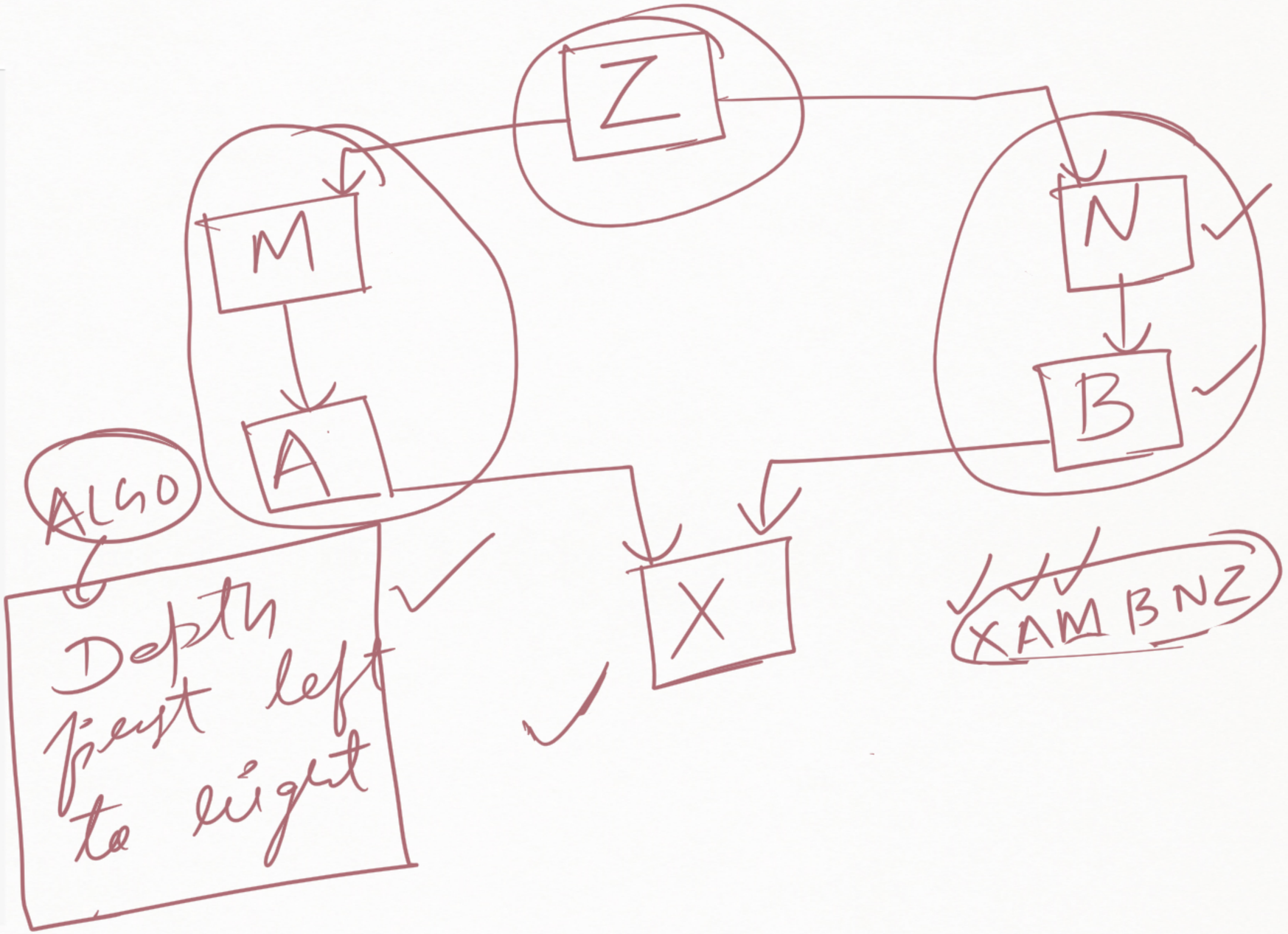
Super() Keyword

→ Calls the immediate MRO method.

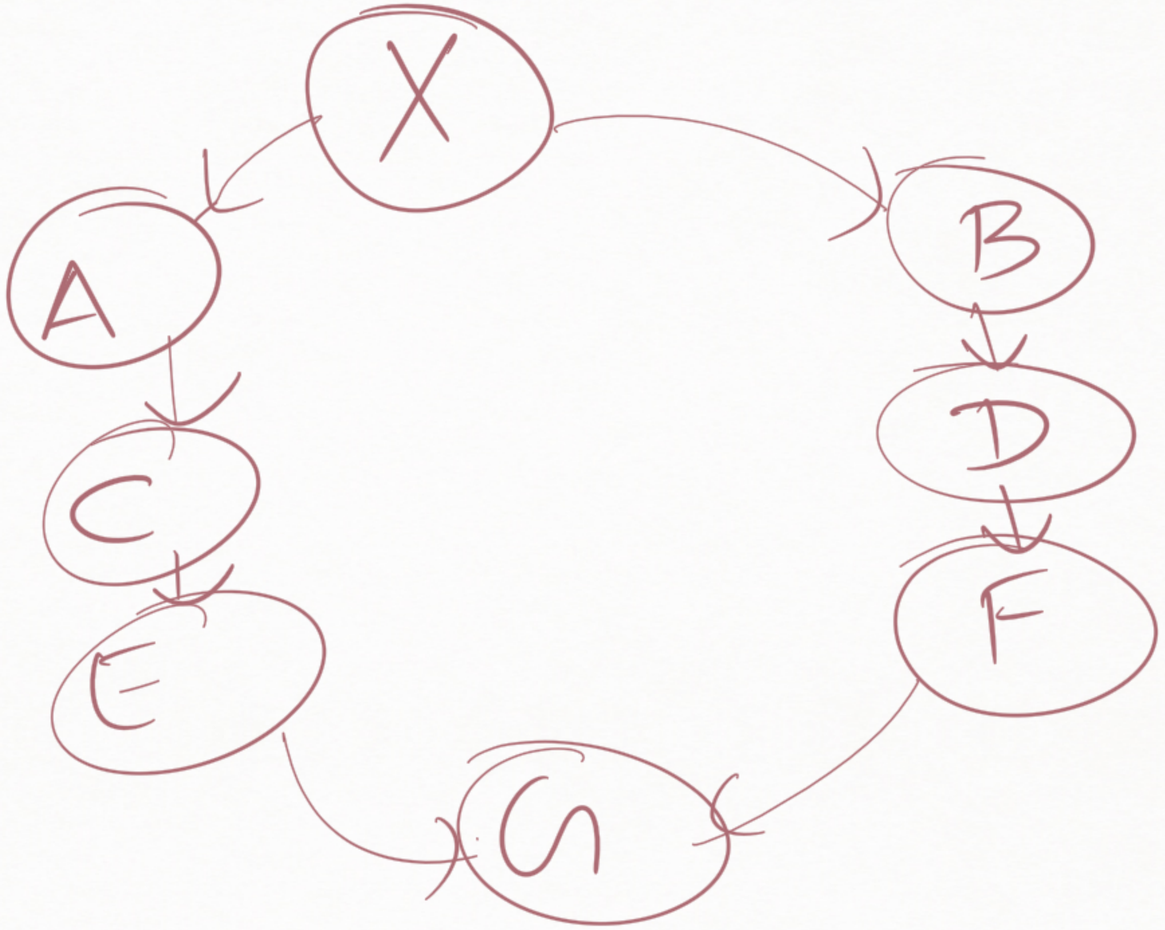
→ Follows the MRO until it finds the method.

In [ ]:

```
class Z:
    def __init__(self):
        print("inside Z")

    def fun(self):
        print("fun of Z")

class M(Z):
    def __init__(self):
        print("inside M")

    def fun(self):
        print("fun of M")

class N(Z):
    def __init__(self):
        print("inside N")

    def fun(self):
        print("fun of N")

class A(M):
    def __init__(self):
        print("inside A")

    def fun(self):
        print("fun of A")

class B(N):
    def __init__(self):
        print("inside B")

    def fun(self):
        print("fun of B")

class X(A,B):
    def __init__(self):
        print("inside X")

    def fun(self):
        print("fun of X")
```



Depth
IN
Left to
Right



MRO = GECAFDBX

In [ ]:
```
## Access Specifiers

-> By default var and methods are public

-> Private: (__varname) (you should only access these inside the class) [Double underscore]

-> Protected: (_varname) (These should be accessed inside the inherited clssess as well!!) [Single underscore]
```

In [56]:
```
class A:
    def eatmethod(self):
        self.__eat = 3

class B(A):
    def happy(self):
        self.eatmethod()
        print(self.__eat)
        print("happy after eating")
```