```python
class A:
    x = 'class A'
class B(A):
    x = 'class B'
class C(B):
    x = 'class C'
```

class A:
    x = 'class A'

class B(A):
    x = ~~class~~ A'
    x = 'class B'

class C(B):
    x = ~~class~~ B'
    x = 'class C'

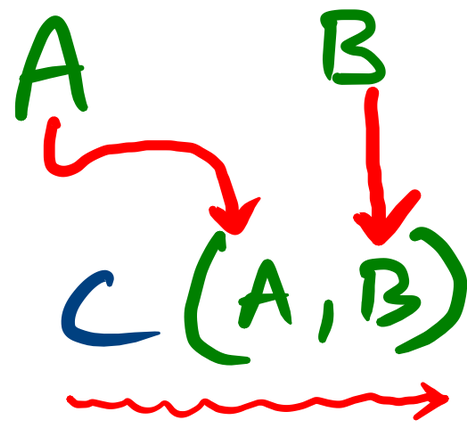
Multiple-Inheritence

```
class A:
    x = 'class A'
class B:
    x = 'class B'
class C(A,B):
    x = 'class C'
```

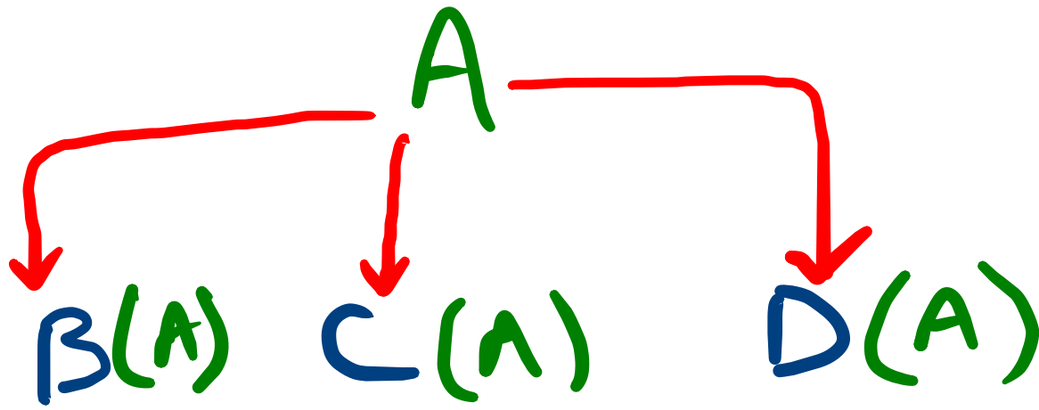
class A :
    x = 'class A'

class B:
    x = 'class B'

class C(A, B):
    x = 'class B'
    x = 'class A'
    x = 'class C'



Hirarchical Inheritence

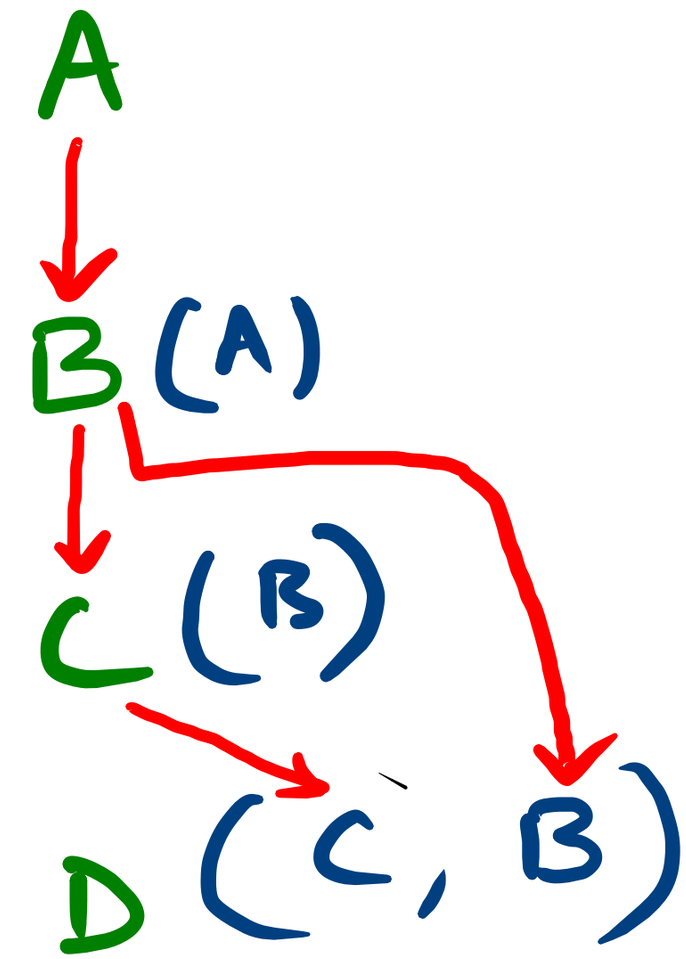



```
class A:
    x = 'class A'
class B(A):
    x = 'class B'
class C(A):
    x = 'class C'
class D(A):
    x = 'class D'
```

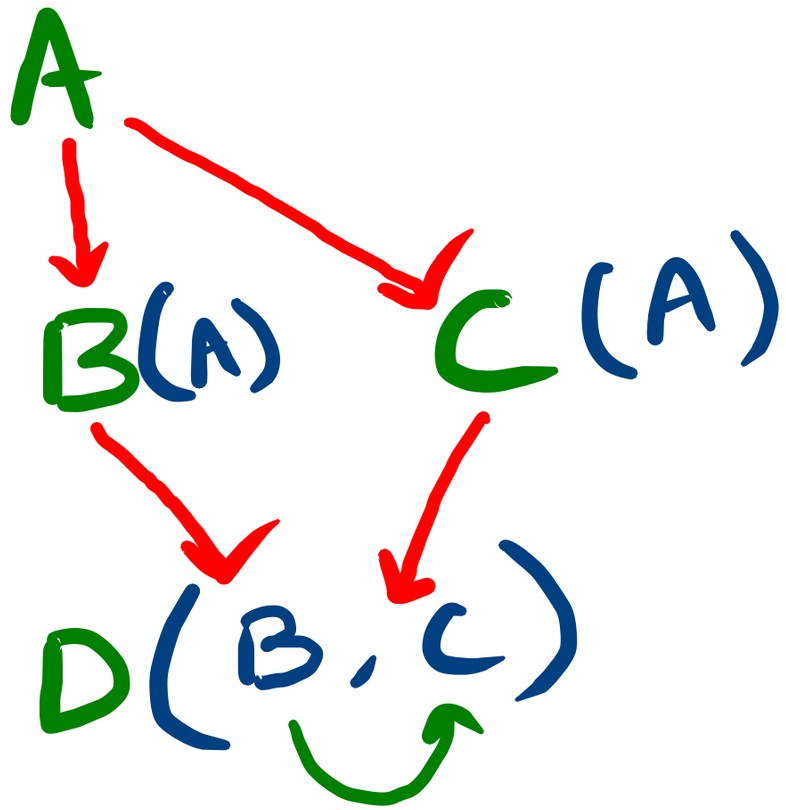
Hybrid Inheritence

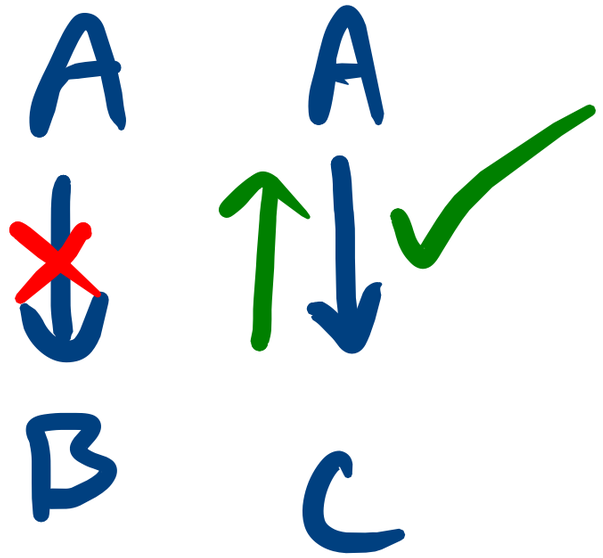
```
class A:
    x = 'class A'
class B(A):
    x = 'class B'
class C(B):
    x = 'class C'
class D(C,B):
    x = 'class D'

print(D.x)
```

```
class D
```



D → C → B → A





D → B → A → C → A ✗

D → B → A ✗

D → B → C → A ✓

D → B → A → C ✗

MRO : Method Resolution order