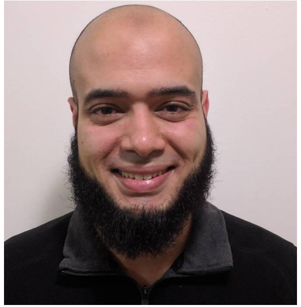
Python Programming

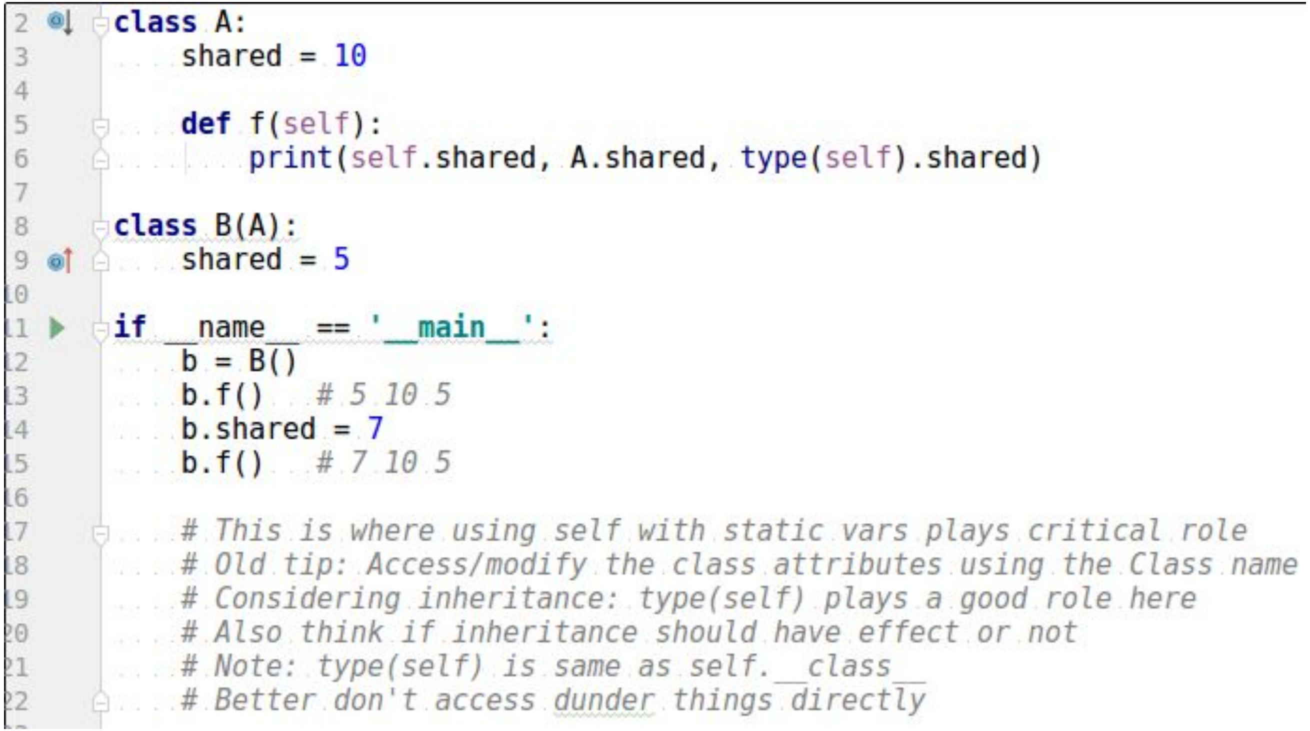
Inheritance with Static vars

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



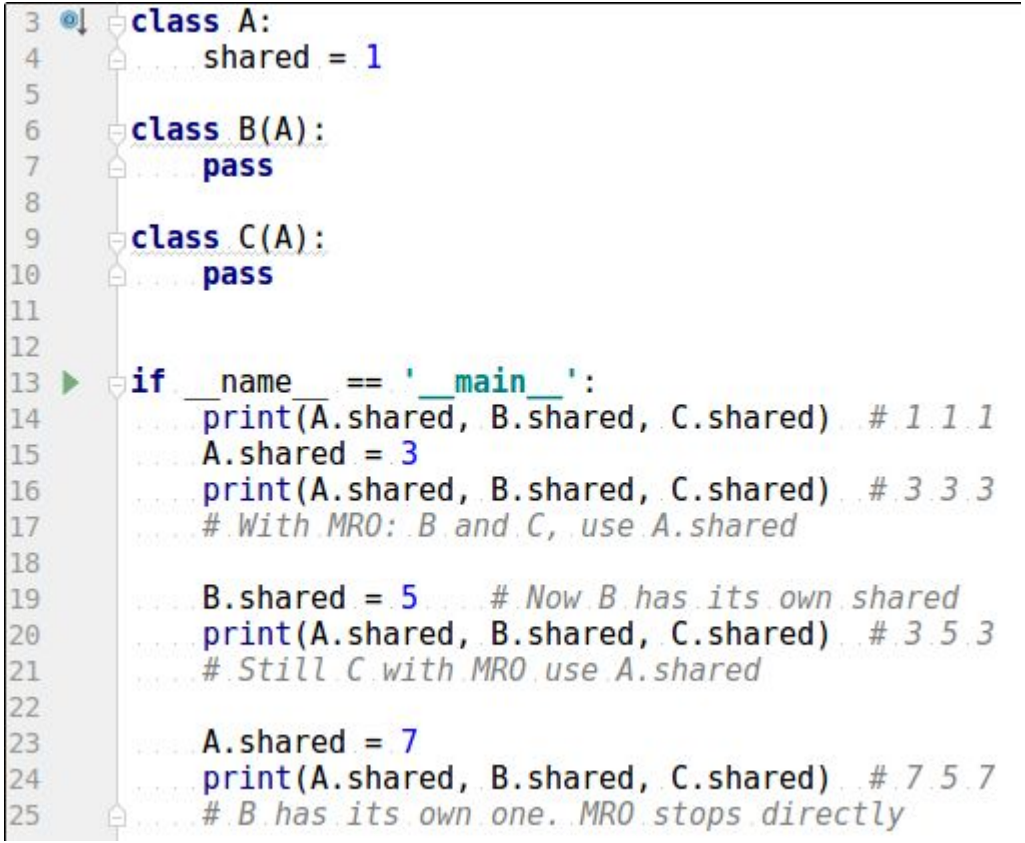
Static variables!



```python
class A:
    shared = 10

    def f(self):
        print(self.shared, A.shared, type(self).shared)

class B(A):
    shared = 5

if __name__ == '__main__':
    b = B()
    b.f()   # 5 10 5
    b.shared = 7
    b.f()   # 7 10 5

    # This is where using self with static vars plays critical role
    # Old tip: Access/modify the class attributes using the Class name
    # Considering inheritance: type(self) plays a good role here
    # Also think if inheritance should have effect or not
    # Note: type(self) is same as self.__class__
    # Better don't access dunder things directly
```

MRO!

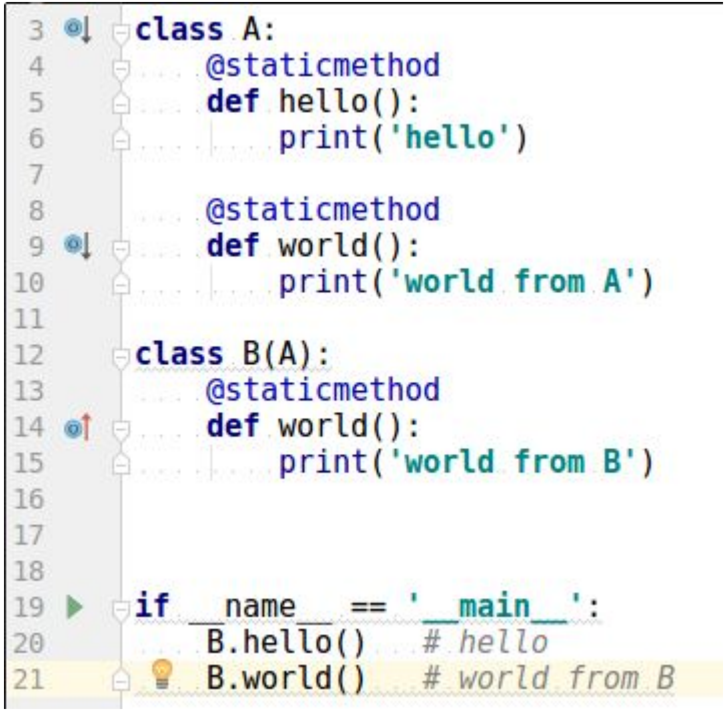
- When changing a static variable, you have to remember the MRO rules and what we learned about static vars



```python
class A:
    shared = 1

class B(A):
    pass

class C(A):
    pass


if __name__ == '__main__':
    print(A.shared, B.shared, C.shared)  # 1 1 1
    A.shared = 3
    print(A.shared, B.shared, C.shared)  # 3 3 3
    # With MRO: B and C, use A.shared

    B.shared = 5    # Now B has its own shared
    print(A.shared, B.shared, C.shared)  # 3 5 3
    # Still C with MRO use A.shared

    A.shared = 7
    print(A.shared, B.shared, C.shared)  # 7 5 7
    # B has its own one. MRO stops directly
```

With static methods

- Intuitive, nothing specific



```
class A:
    @staticmethod
    def hello():
        print('hello')

    @staticmethod
    def world():
        print('world from A')

class B(A):
    @staticmethod
    def world():
        print('world from B')



if __name__ == '__main__':
    B.hello()   # hello
    B.world()   # world from B
```

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”