In [6]:

```
# Example 01
class Boy:
    name =""
    gender ="Male"
    age = 20

class Girl:
    name =""
    gender ="Female"
    age = 18

Ali = Boy()
Aqsa = Girl()

print('My name is Ali')
print(Ali.gender)
Ali.age = 50
print(Ali.age)

print('My name is Aqsa')
print(Aqsa.gender)
Aqsa.age = 20
print(Aqsa.age)
```

```
My name is Ali
Male
50
My name is Aqsa
Female
20
```

In [22]:

```
# Example 02
class POS:
    count = 0
    def __init__(self):
        type(self).count += 1
    def __del__(self):
        type(self).count -= 1

if __name__ == '__main__':
    x1 = POS()
    print("Number of instances : "+ str(POS.count))
    x2 = POS()
    print("Number of instance :" + str(POS.count))
    x3 = POS()
    print("Number of instance :" + str(POS.count))
    del x2
    print("Number of instance :" + str(POS.count))
    del x1
    print("Number of instance :" + str(POS.count))
```

```
Number of instances : 1
Number of instance :2
Number of instance :3
Number of instance :2
Number of instance :1
```

In [25]:

```
# Example 03
class Fish:
    def __init__(self, f_name, l_name='Fish'):
        self.f_name = f_name
        self.l_name = l_name
    def swim(self):
        print(f'{self.f_name} Fish is swimming.')
    def swim_back(self):
        print(f'{self.f_name} Fish can swim backwards.')

nimo = Fish('Nimo')
nimo.swim()
nimo.swim_back()
```

```
Nimo Fish is swimming.
Nimo Fish can swim backwards.
```

In [26]:

```python
# Example 04
class Fish:
    def __init__(self, f_name, l_name='Fish', skeleton='Bone', eye_lids=False):
        self.f_name = f_name
        self.l_name = l_name
        self.skeleton = skeleton
        self.eye_lids = eye_lids
    def swim(self):
        print(f'{self.f_name} Fish is swimming.')
    def swim_back(self):
        print(f'{self.f_name} Fish can swim backwards.')

nimo = Fish('Nimo', skeleton='cartilage', eye_lids=True)
nimo.swim()
nimo.swim_back()
```

```
Nimo Fish is swimming.
Nimo Fish can swim backwards.
```

In [31]:

```python
# Example 05
class Trout(Fish):
    pass
terry = Trout(f_name='Terry', skeleton='Cartilage', eye_lids=True)
print(terry.f_name, terry.l_name)
print(terry.skeleton)
print(terry.eye_lids)
terry.swim()
terry.swim_back()
```

```
Terry Fish
Cartilage
True
Terry Fish is swimming.
Terry Fish can swim backwards.
```

In [34]:

```python
# Example 06
class ClownFish(Fish):
    def live(self):
        print(f'{self.f_name}Fish is coexisting with sea anemone.')

clown = ClownFish(f_name='Clown')
clown.swim()
clown.swim_back()
clown.live()
```

```
Clown Fish is swimming.
Clown Fish can swim backwards.
ClownFish is coexisting with sea anemone.
```

In [36]:

```
# Example 07
class Shark(Fish):
    def init (self, f_name, l_name="Shark", skeleton=None, eye_lids=True):
        self.f_name = f_name
        self.l_name = l_name
        self.skeleton = skeleton
        self.eye_lids = eye_lids
    def swim_back(self):
        print(f'{self.f_name} shark cannot swim backwards, but can sink backwards.')

sammy = Shark(f_name='Sammy', skeleton='Cartilage')
print(sammy.f_name, sammy.l_name)
print(sammy.skeleton)
print(sammy.eye_lids)
sammy.swim()
sammy.swim_back()          # Overridden Method
```

```
Sammy Fish
Cartilage
False
Sammy Fish is swimming.
Sammy shark cannot swim backwards, but can sink backwards.
```

In [38]:

```
# Example 08
class Trout(Fish):
    def __init__(self, water='FreshWater'):
        self.water = water
        super().__init__(self)

terry = Trout()
terry.f_name = 'Terry'
print(terry.f_name, terry.l_name)
print(terry.skeleton)
print(terry.eye_lids)
print(terry.water)
terry.swim()
```

```
Terry Fish
Bone
False
FreshWater
Terry Fish is swimming.
```

In [39]:

```
# Example 09
class Coral:
    def community(self):
        print("Coral lives in a community.")

class Anemone:
    def protect_clownfish(self):
        print("The anemone is protecting the clownfish.")

class CoralReef(Coral, Anemone):
    pass

barrier = CoralReef()
barrier.community()
barrier.protect_clownfish()
```

```
Coral lives in a community.
The anemone is protecting the clownfish.
```

In [61]:

```
# Task 02
class Passenger_Plane:
    def __init__(self, name, flight_num, capacity):
        self.name = name
        self.flight_num = flight_num
        self.capacity = capacity
    def get_details(self):
        print(f'Flight Name: {self.name}\nFlight Number: {self.flight_num}\nCapacity:
{self.capacity}')
    def book_seat(self):
        self.seat = True

class Fighter_Plane:
    def __init__(self, name):
        self.name = name
    def get_name(self):
        return self.name
    def isfighter(self):
        return True

class Plane(Passenger_Plane, Fighter_Plane):
    def __init__(self, name, flight_num=None, capacity=None):
        Passenger_Plane.__init__(self, name, flight_num, capacity)
    def get_details(self):
        super().get_details()
```

In [62]:

```
pia = Plane('PIA', 'PK-123', 300)
pia.get_details()

jf = Plane('JF-17 Thunder')
print(jf.get_name())
print('It is Fighter:', jf.isfighter())
```

```
Flight Name: PIA
Flight Number: PK-123
Capacity: 300
JF-17 Thunder
It is Fighter: True
```

In [84]:

```
# Task 03
class Umbrella:
    def __init__(self, use, color, size, material, price):
        self.use = use
        self.color = color
        self.size = size
        self.material = material
        self.price = price
    def get_price(self):
        return self.price

class Rain_Umbrella(Umbrella):
    def __init__(self, use, color, size, material, price):
        Umbrella.__init__(self, use, color, size, material, price)
    def get_use(self):
        return self.use

class Sun_Umbrella(Rain_Umbrella, Umbrella):
    def __init__(self, use, color, size, material, price):
        super().__init__(use, color, size, material, price)
    def get_materialUsed(self):
        return self.material
```

In [87]:

```
sun = Sun_Umbrella(use='Sunlight', color='Black', size='Medium',material='Fabrique', pr
ice=100)
print(sun.get_price())
print(sun.get_use())
print('Used Material:', sun.get_materialUsed())
```

```
100
Sunlight
Used Material: Fabrique
```