

```
In [1]: ▶ 1 # Exercise 01
2 from abc import ABC, abstractmethod
3 class NetworkInterface(ABC):
4     @abstractmethod
5     def connect(self):
6         pass
7     @abstractmethod
8     def transfer(self):
9         pass
10
11 class RealNetwork(NetworkInterface):
12     def connect(self):
13         print("Network is connect")
14     def transfer(self):
15         print("All Data is tranfser")
16
17 R1= RealNetwork()
18 R1.connect()
19 R1.transfer()
```

executed in 19ms, finished 23:08:34 2020-07-18

Network is connect  
All Data is tranfser

In [2]:

```
1 # Exercise 02
2 from abc import ABC, abstractmethod
3 class NetworkInterface(ABC):
4     @abstractmethod
5     def connect(self):
6         pass
7     @abstractmethod
8     def transfer(self):
9         pass
10
11 class RealNetwork(NetworkInterface):
12     def connect(self):
13         print("Network is connect")
14     def transfer(self):
15         print("All Data is tranfser")
16
17 class FakeNetwork(NetworkInterface):
18     def connect(self):
19         print("Network is connect")
20
21     def transfer(self):
22         print("All Data is tranfser")
23
24 R1= RealNetwork()
25 R1.connect()
26 R1.transfer()
27 F1= FakeNetwork()
28 F1.connect()
```

executed in 144ms, finished 23:08:34 2020-07-18

```
Network is connect
All Data is tranfser
Network is connect
```

In [3]:

```
1 # Exercise 03
2 class A:
3     def __init__(self, x,y):
4         self.x = x
5         self.y = y
6
7     def __add__(self, other):
8         x = self.x + other.x
9         y = self.y + other.y
10        return(x,y)
11
12 c1 = A(5,3)
13 c2 = A(2,3)
14 print("Sum:", c1+c2)
```

executed in 183ms, finished 23:08:34 2020-07-18

```
Sum: (7, 6)
```

In [4]:

```
1 # Exercise 04
2 class A:
3     def __init__(self, x):
4         self.x = x
5
6     def __lt__(self, other):
7         if self.x < other.y:
8             return True
9         else:
10            return False
11
12 class B:
13     def __init__(self, y):
14         self.y = y
15
16 c1 = A(2)
17 c2 = B(4)
18 print("Is c1 less than c2: ",c1<c2)
```

executed in 146ms, finished 23:08:34 2020-07-18

Is c1 less than c2: True

In [5]:

```
1 # Exercise 05
2 class A:
3     def __init__(self, a):
4         self.a = a
5
6     def __lt__(self, other):
7         if self.a < other.a:
8             return True
9         else:
10            return False
11
12     def __eq__(self, other):
13         if(self.a == other.a):
14             return "Both are equal"
15         else:
16             return "Not equal"
17
18 ob1 = A(2)
19 ob2 = A(3)
20 print(ob1 < ob2)
21
22 ob3 = A(4)
23 ob4 = A(4)
24 print(ob3 == ob4)
```

executed in 111ms, finished 23:08:34 2020-07-18

True

Both are equal

In [6]:



```
1 # Task 01
2 class A:
3     def __init__(self, x):
4         self.x = x
5
6     def __gt__(self, other):
7         if self.x > other.y:
8             return True
9
10        else:
11            return False
12
13    def __lt__(self, other):
14        if self.x < other.y:
15            return True
16
17        else:
18            return False
19
20    def __ge__(self, other):
21        if self.x >= other.y:
22            return True
23
24        else:
25            return False
26
27    def __le__(self, other):
28        if self.x <= other.y:
29            return True
30
31        else:
32            return False
33
34    def __eq__(self, other):
35        if(self.x == other.y):
36            return True
37        else:
38            return False
39
40    def __ne__(self, other):
41        if(self.x != other.y):
42            return "Both are equal"
43        else:
44            return "Not equal"
45
46 class B:
47     def __init__(self, y):
48         self.y = y
```

executed in 113ms, finished 23:08:35 2020-07-18

```
In [7]: ▶ 1 num1 = A(2)
2 num2 = B(4)
3 print("Is num1 less than num2? ", num1 > num2)
4
5 num1 = A(2)
6 num2 = B(2)
7 print("Is num1 is equal to num2? ", num1 == num2)
8
9 num1 = A(8)
10 num2 = B(4)
11 print("Is num1 is less than or equal to num2? ", num1 <= num2)
```

executed in 117ms, finished 23:08:35 2020-07-18

Is num1 less than num2? False  
Is num1 is equal to num2? True  
Is num1 is less than or equal to num2? False

```
In [8]: ▶ 1 # Task 02 Robots Example
2 from abc import ABC, abstractmethod
3 # Abstract Class
4 class Robot(ABC):
5     @abstractmethod
6     def obeyOrder(self): pass
7
8     @abstractmethod
9     def doCleaning(self): pass
10
11 # Derived Classes
12 class Cook(Robot):
13     def obeyOrder(self):
14         print('Cook Robot is cooking.')
15     def doCleaning(self):
16         print('Cook Robot is cleaning kitchen.')
17     def cooking(self, dish):
18         print(f'Robot is cooking {dish}.')
19     def baking(self):
20         print('Robot is baking cookies.')
21
22 class Driver(Robot):
23     def obeyOrder(self):
24         print('Robot is driving.')
25     def doCleaning(self):
26         print('Robot is cleaning the vehicle.')
27     def drive(self, speed_status):
28         print(f'Robot is driving {speed_status}.')
29     def driveTo(self, destination):
30         print(f'Robot is driving to {destination}')
```

executed in 173ms, finished 23:08:35 2020-07-18

In [9]:



```
1 cook = Cook()
2 cook.obeyOrder()
3 cook.cooking('fish')
4
5 driver = Driver()
6 driver.doCleaning()
7 driver.driveTo('Clifton')
```

executed in 104ms, finished 23:08:35 2020-07-18

Cook Robot is cooking.  
Robot is cooking fish.  
Robot is cleaning the vehicle.  
Robot is driving to Clifton