Inheritence : Transfering one class attributes or members to other class

```
class class1:
  x = 10

class class2:
  y = 20

print(class1.x)
print(class2.y)
print(class2.x)
```

```
10
20
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-1-cd36ac1af4ad> in <cell line: 9>()
      7 print(class1.x)
      8 print(class2.y)
----> 9 print(class2.x)

AttributeError: type object 'class2' has no attribute 'x'
```

SEARCH STACK OVERFLOW

```
class class1:
  x = 10

class class2(class1):
  y = 20

print(class1.x)
print(class2.y)
print(class2.x)
```

```
10
20
10
```

single inheritence

```
class class1:
  x = 10
```

```
class class2(class1):
  x = 20

print(class2.x)
```

```
20
```

```
class class1:
  x = 10

class class2(class1):
  pass

print(class1.x)
print(class2.x)
```

```
10
10
```

```
class class1:
  x = 'python'
  @classmethod
  def showdata(cls):
    print(cls.x)
class class2:
  x = 'django'

print(class1.x)
print(class2.x)
```

```
python
django
```

```
class1.showdata()
```

```
python
```

```
class2.showdata()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-11-39d37efe29bc> in <cell line: 1>()
----> 1 class2.showdata()

AttributeError: type object 'class2' has no attribute 'showdata'
```

SEARCH STACK OVERFLOW

```
class class1:
```

```
    x = 'python'
    @classmethod
    def showdata(cls):
      print(cls.x)
class class2(class1):
    x = 'django'

print(class1.x)
print(class2.x)
```

```
python
django
```

```
class1.showdata()
```

```
python
```

```
class2.showdata()
```

```
django
```

```
class class1:
  def __init__(self):
    print('class : class1')

class class2(class1):
  pass

c11 = class1()
```

```
class : class1
```

```
c21 = class2()
```

```
class : class1
```

```
print(type(c11).__name__)
```

```
class1
```

```
print(type(c21).__name__)
```

```
class2
```

```
class class1:
  def __init__(self):
    print('class : class1')
```

```
class class2(class1):
  def __init__(self):
    print('class : class2')

c11 = class1()
```

```
class : class1
```

```
c21 = class2()
```

```
class : class2
```

```
print(type(c11).__name__)
```

```
class1
```

```
print(type(c21).__name__)
```

```
class2
```

```
class mobile_v1:
  os = 'symbion'
  touch = 'not supported'
  keypad = 'available'
  internet = 'not supported'
  radio = 'available'
  camera = 'not supported'
  @classmethod
  def showdata(cls):
    print('Operating System :',cls.os)
    print('Touch screen :',cls.touch)
    print('Keypad :',cls.keypad)
    print('Internet :',cls.internet)
    print('Radio :',cls.radio)
    print('Camera :',cls.camera)

class mobile_v2(mobile_v1):
  os = 'android'
  touch = 'available'
  keypad = 'not supported'
  internet = 'available'
  camera = 'available'

mobile_v1.showdata()
```

```
Operating System : symbion
Touch screen : not supported
Keypad : available
```

```
Internet : not supported
Radio : available
Camera : not supported
```

```
mobile_v2.showdata()
```

```
Operating System : android
Touch screen : available
Keypad : not supported
Internet : available
Radio : available
Camera : available
```

✓ 0s completed at 3:09 PM