In [42]:
```
class Employee:

    employeeCounter = 0;

    def __init__(self,firstName,lastName):
        self.firstName = firstName
        self.lastName = lastName
        Employee.employeeCounter += 1
        print(" hello i employee")


    def displayEmployeeName(self):
        print(self.firstName + " " + self.lastName)

    def displayHello(self):
        print("hello from Employee class")
```

In [3]:
```
emp = Employee("ahmed","khalifa")
```

```
 hello i employee
```

In [4]:
```
print(Employee.employeeCounter)
```

```
1
```

In [5]:
```
emp.displayEmployeeName()
```

```
ahmed khalifa
```

ex_01.Inheritance

In [6]:
```
class Developer(Employee):
    pass
```

1.1 take care with constructor parameters of parent

In [7]:
```
dev = Developer()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-7-941e249cd05a> in <module>()
----> 1 dev = Developer()

TypeError: __init__() missing 2 required positional arguments: 'firstName' an
d 'lastName'
```

In [8]:
```
dev = Developer("ahmed","khalifa")
```

```
 hello i employee
```

1.2 working with class variable in Inheritance

In [9]:
```
print(Employee.employeeCounter)
```

```
2
```

In [10]:
```
print(Developer.employeeCounter)
```

```
2
```

In [11]:
```
Developer.employeeCounter = 10
```

In [12]:
```
print(Developer.employeeCounter)
```

```
10
```

In [13]:
```
print(Employee.employeeCounter)
```

```
2
```

1.3 working with instance variable in Inheritance

In [14]:
```
print(dev.firstName)
```

```
ahmed
```

1.4 working with Methods in Inheritance

In [15]:
```
dev.displayEmployeeName()
```

```
ahmed khalifa
```

ex_02.Inheritance

In [43]:
```
class SoftwareEngineer(Employee):
    employeeCounter = 0
    def __init__(self, fullName,department):
        self.fullName = fullName
        self.department = department
        SoftwareEngineer.employeeCounter += 1
        print(" i am software engineer ")
    def displaySoftwareEngName(self):
        print(self.fullName + " worked at " + self.department )
```

2.1 take care with constructor parameters

In [44]:
```
sw = SoftwareEngineer("ahmed khalifa","web")
```

```
 i am software engineer
```

In [18]:
```
print(SoftwareEngineer.employeeCounter)
```

```
1
```

In [19]:
```
print(Employee.employeeCounter)
```

```
2
```

In [20]:
```
emp3 = Employee("a","z")
```

```
 hello i employee
```

In [21]:
```
print(SoftwareEngineer.employeeCounter)
```

```
1
```

In [22]:
```
print(Employee.employeeCounter)
```

```
3
```

In [23]:
```
print(Developer.employeeCounter)
```

```
10
```

In [24]:
```
dev2 = Developer("zxc" , "asd")
```

```
 hello i employee
```

In [25]:
```
print(Employee.employeeCounter)
print(Developer.employeeCounter)
print(SoftwareEngineer.employeeCounter)
```

```
4
10
1
```

2.3 working with instance variable in Inheritance

In [31]:
```
print(sw.fullName)
```

```
ahmed khalifa
```

In [32]:
```
print(sw.firstName)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-32-bc14e0877824> in <module>()
----> 1 print(sw.firstName)

AttributeError: 'SoftwareEngineer' object has no attribute 'firstName'
```

2.4 working with Methods in Inheritance

In [33]:
```
sw.displaySoftwareEngName()
```

```
ahmed khalifa worked at web
```

In [34]:
```
sw.displayEmployeeName()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-34-becaad5c5659> in <module>()
----> 1 sw.displayEmployeeName()

<ipython-input-2-9e380b9cb2c5> in displayEmployeeName(self)
     11
     12     def displayEmployeeName(self):
---> 13         print(self.firstName + " " + self.lastName)
     14

AttributeError: 'SoftwareEngineer' object has no attribute 'firstName'
```

In [45]:
```
sw.displayHello()
```

```
hello from Employee class
```

ex_03_Inheritance

In [73]:
```
class Desinger(Employee):
     def __init__(self,age,firstName,department,lastName):
        self.age = age
        self.firstName = firstName
        self.department = department
        self.lastName = lastName
        print(" hello i designer")
```

In [75]:
```
d = Desinger(25,'lala','photoshop','nana')
```

```
 hello i designer
```

In [76]:
```
d.displayEmployeeName()
```

```
lala nana
```

In [ ]:
```
```