(7.6) Access Specifier, Name Mangling

Access Specifiers

By default every attribute is public

Public Scope

we can access and manipluate public variables and methods from outside the class through instance

Protected Scope

Protected attributes can be accessed within the class anywhere but from outside of the class only in child classes. We can specify an attribute as protected by prefexing with _ symbol

_variablename = value

But it is just convention and in reality does not exists protected attributes

Private Scope

can not be accessed or manipluated from outside the class, only methods can access and manipluate them
generally used to hide some sensiteive information from outside the class
We can declare a variable as private explicitly by prefexing with 2 underscore symbols.

__variablename = value

In [1]:
```
class Test:
    x=10
    _y=20
    __z=30
    def m1(self):
        print(Test.x)
        print(Test._y)
        print(Test.__z)

t=Test()
t.m1()
print(Test.x)
print(Test._y)
print(Test.__z)
```

```
10
20
30
10
20
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
C:\Users\PANKAJ~1\AppData\Local\Temp/ipykernel_360/2597779358.py in <module>
     12 print(Test.x)
```

```
     13 print(Test._y)
---> 14 print(Test.__z)

AttributeError: type object 'Test' has no attribute '__z'
```

Name Mangling

python automatically change the name of __variables into _cls__variable to make them private
so that we can not directly / accediently access them from outside the class

data --> _ Adata

In [3]:
```
class Test:
    def __init__(self):
        self.__x=10

t=Test()
print(t._Test__x)#10
```

```
10
```

Problems On Name Mangling

In [10]:
```
class A:
    __msg = ""
    def __init__(self):
        self.__secret = 'Some Secret That Should not accessed outside the class'
        # Private variable names changed
        # into _cls__variable name, _A__secret
    def get_secret(self):
        return self.__secret
    def update_secret(self, new_value):
        self.__secret = new_value
```

In [11]:
```
A.__msg # private class variable
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
C:\Users\PANKAJ~1\AppData\Local\Temp/ipykernel_360/2097255600.py in <module>
----> 1 A.__msg # private class variable

AttributeError: type object 'A' has no attribute '__msg'
```

In [12]:
```
#but also we can access this as
A._A__msg
```

Out[12]:
```
''
```

In [13]:
```
a = A()
```

In [14]:
```
a.__secret # Data Hiding can not be access directly
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
C:\Users\PANKAJ~1\AppData\Local\Temp/ipykernel_360/2077942877.py in <module>
----> 1 a.__secret # Data Hiding can not be access directly
```

```
AttributeError: 'A' object has no attribute '__secret'
```

In [15]:
```
a.get_secret() # can be access through object method
```

Out[15]:
```
'Some Secret That Should not accessed outside the class'
```

In [16]:
```
a.update_secret('ha ha ha updated')
```

In [17]:
```
a.get_secret()
```

Out[17]:
```
'ha ha ha updated'
```

In [18]:
```
a.__secret
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
C:\Users\PANKAJ~1\AppData\Local\Temp/ipykernel_360/409297407.py in <module>
----> 1 a.__secret

AttributeError: 'A' object has no attribute '__secret'
```

In [19]:
```
a.__secret = 'yahooo i have changed it.'
# this is another object variable
```

In [20]:
```
a.__secret
```

Out[20]:
```
'yahooo i have changed it.'
```

In [21]:
```
a.get_secret()
```

Out[21]:
```
'ha ha ha updated'
```

In [22]:
```
a._A__secret # even we can access this as
```

Out[22]:
```
'ha ha ha updated'
```

In [23]:
```
a.get_secret() # self._A__var -> instance, cls._A__var
```

Out[23]:
```
'ha ha ha updated'
```

In [24]:
```
print(a.__dict__)

# _A__secret =
```

```
{'_A__secret': 'ha ha ha updated', '__secret': 'yahooo i have changed it.'}
```

Private Methods

In [26]:
```
class A:
```

```
    def __some_method(self):
        print("Hey Can you see me outside !!")
    @classmethod
    def __some_cls(cls):
        print("Secret / Private Class Space")

    def a(self):
        self.__some_method()
    def b(self):
        self.__some_cls()
```

In [28]:

```
a = A()
a.a()
a.b()
```

```
Hey Can you see me outside !!
Secret / Private Class Space
```

In [29]:

```
a.__some_method()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
C:\Users\PANKAJ~1\AppData\Local\Temp/ipykernel_360/3029436480.py in <module>
----> 1 a.__some_method()

AttributeError: 'A' object has no attribute '__some_method'
```

In [30]:

```
a.__some_cls()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
C:\Users\PANKAJ~1\AppData\Local\Temp/ipykernel_360/45259027.py in <module>
----> 1 a.__some_cls()

AttributeError: 'A' object has no attribute '__some_cls'
```

In [31]:

```
A.__some_cls()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
C:\Users\PANKAJ~1\AppData\Local\Temp/ipykernel_360/3580010948.py in <module>
----> 1 A.__some_cls()

AttributeError: type object 'A' has no attribute '__some_cls'
```

In [ ]: