Encapsulation_[Day_13]_(DATA_MINDS)

September 14, 2023

ENCAPSULATION - IN ENCAPSULATION WE SIMPLY WRAP THE VARIABLE OR ENCAPSULATE THE VARIABLE THAT DIRECTLY DOES'NT SEEN BY THE USER , THEY CAN SEEN ONLY THAT PART OF CODE THAT WE WANT TO PRESENT IN FORNT OF HIM/HER.

USER CAN NOT SEEN THE INTERNAL ACTUAL COMPOSITION OF CLASS , USER CAN ONLY MODIFY THE DATA INSIDE CLASS THAT WE WANT TO GIVE HIM OR SHOW HIM

[1]:
```
class test:
    def __init__(self,a,b):
        self.a=a
        self.b=b
```

[2]:
```
t=test(10,25)
```

[3]:
```
t.a=1235
```

[4]:
```
t.a
```

[4]: 1235

[14]:
```
#WITH THE HELP OF DOUBLE UNDERSCORE( __ ) WE MAKE THE VARIABLE PRIVATE THAT CAN␣
 ↪NOT BE ACCESS DIRECTLTY
#WE SIMPLY USE DOUBLE UNDERSCORE( __ ) WITH SELF AND WHENEVER WE ACCESS THE␣
 ↪VARIABLE OR ELEMENT AT THAT TIME WE ALSO USE THE DOUBLE UNDRERSCORE( __ )

class car:
    def __init__(self,year,make,model,speed):
        self.__year=year
        self.__make=make
        self.__model=model
        self.__speed=0
    def set_speed(self,speed):
        self._speed=0 if speed<0 else speed
    def get_speed(self):
        return self.__speed
```

[15]:
```
c=car(2023,"BMW","BMW X6 Sport",250)
```

[16]:
```
c._car__year
```

[16]:
```
2023
```

[17]:
```
c._car__model
```

[17]:
```
'BMW X6 Sport'
```

[18]:
```
c._car__speed
```

[18]:
```
0
```

[19]:
```
c._car__make
```

[19]:
```
'BMW'
```

[23]:
```
c.set_speed(-100)
```

[26]:
```
c.get_speed()
```

[26]:
```
0
```

DEMOSTRATION OF BANK ACCOUNT WORKING -:

IN THIS SIMPLE CODE WE DEPOSIT , WITHDRAW AND CHECK THE BANK BALANCE AND WE ALSO DONE SAME FUCTIONS THAT WE DONE IN BANKS.

[40]:
```
class bank_account:
    def __init__(self,balance):
        self.__balance=balance
    def deposit(self,amount):
        self.__balance=self.__balance+amount
    def withdraw(self,amount):
        if self.__balance>=amount:
            self.__balance=self.__balance-amount
            return True
        else:
            return False
    def get_balance(self):
        return self.__balance
```

[41]:
```
Virat=bank_account(1000)
```

[42]:
```
Virat.get_balance()
```

[42]:
```
1000
```

[44]:
```
Virat.withdraw(500)
```

[44]: True

[45]:
```
Virat.get_balance()
```

[45]: 500

[46]:
```
Virat.deposit(2500)
```

[48]:
```
Virat.get_balance()
```

[48]: 3000

[49]:
```
Virat.withdraw(6000)
```

[49]: False

[50]:
```
Virat.deposit(15000)
```

[51]:
```
Virat.get_balance()
```

[51]: 18000

[52]:
```
Virat.withdraw(5000)
```

[52]: True

[53]:
```
Virat.get_balance()
```

[53]: 13000