

encapsulation eng

February 12, 2023

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

```
[3]: t = test(45,56)
```

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

```
[4]: 45
```

```
[5]: t.b
```

```
[5]: 56
```

```
[6]: t.a = 3243
```

```
[7]: t.a
```

```
[7]: 3243
```

```
[27]: 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
```

```
[29]: obj_car.get_speed()
```

```
[29]: 0
```

```
[20]: obj_car.set_speed(-1234)
```

```
[21]: obj_car.set_speed(3453)
```

```
[28]: obj_car = car(2021 , "toyota" , "innova" , 12)
```

```
[17]: obj_car._car__year = 2023
```

```
[18]: obj_car._car__year
```

```
[18]: 2023
```

```
[22]: obj_car._car__speed
```

```
[22]: 3453
```

```
[ ]: obj_car
```

```
[31]: 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
```

```
[32]: obj_bank_account = bank_account(1000)
```

```
[33]: obj_bank_account.get_balance()
```

```
[33]: 1000
```

```
[34]: obj_bank_account.deposit(6000)
```

```
[35]: obj_bank_account.get_balance()
```

```
[35]: 7000
```

```
[36]: obj_bank_account.withdraw(10000)
```

```
[36]: False
```

```
[37]: obj_bank_account.withdraw(2000)
```

```
[37]: True
```

```
[38]: obj_bank_account.get_balance()
```

```
[38]: 5000
```

```
[ ]:
```