```
# This is the Class:

class Student :

   colage_name ="Tnu Colage"

   name ="Anonymous"


#  This is the Constractor;

   def __init__(self , name, age): # ARGUMENT:


     self.name =name    # Argument pass

     self .age =age     # Argument pass


   def hello(self): # method


      print("Welcome Student :",self.name)

      print("And Age is :", self.age)


def get_marks(self): # Method

    return self.age


#  This  is the Obj.

s1=Student("Ramiz", 21)

print(s1.name, s1.age)


s1.hello()

'''
```

```
'''
class Student:


   def __init__(self, name, marks):
      self.name =name
      self.marks=marks


   def get_avg(self):


       sum =0


       for  i in self.marks:


         sum +=i
         Average = sum / 3
       print("hi", self.name, "your avg score is :",Average)


   @staticmethod
   def Hello():


      print("HELLO")


obj=Student("Ramiz",[100] )


obj.get_avg()
obj.name ="fuck"

obj.get_avg()

obj.Hello()

# ! <<<<<<<<<<<<<<Abstraction CODE >>>>>>>>>>>>>>>>>>>>>> !
class Car:

  def __init__(self):

     self.accelerator=False
     self.brk =False
     self.clutch=False
  def start(self):

    self.clutch = True
    self.accelerator=True
    print("Car Start:")

obj=Car()

obj.start()

# ! <<<<<<<<<<<<-Encapsulation->>>>>>>>>>>>>>>>>> !
class Account:
    def __init__(self, bal, acc):
        self.balance = bal
        self.account_no = acc

    # Debit method
    def debit(self, amount):
        self.balance -= amount
        print("Rs.", amount, "was debited")
        print("Total Balance:", self.get_balance())

    # Credit method
    def credit(self, amount):
        self.balance += amount
        print("Rs.", amount, "was credited")
        print("Total Balance:", self.get_balance())

    # Method to get current balance
    def get_balance(self):
        return self.balance

# Creating an account object
acc1 = Account(10000, 12345)

# Performing transactions
acc1.debit(1000)
```

```
acc1.credit(500)
```