Types of methods:
----

in python we can create a methods in 4-ways,they are

1).Instance methods

2).Class methods

3).Static methods

4).Abstarct methods

Instance methods:
----

we can create a Instance methods dont required any decorator(function) concept.

the instance methods takes the self as a default parameter

in instance methods we can access/use the both class level(static) and Instance level(non-static) data.

we can access the instance methods from outside the class with the help reference variable only.

```
        syntax
        ------
        class classname:
                --------
                --------
                def methodname(self):
                        --------
                        --------
```

class methods:
----

we can create a class methods to required classmethod decorator.

the class methods takes the 'cls' as a default parameter

in class methods we can access the only class level(static) data.

we can access the class-methods from outside the class directly by using classname or reference variable.

```
        syntax
        ------
        class classname:
                -------
                -------
                @classmethod
                def methodname(cls):
                        --------
                        --------
```

Static methods:
----

we can create a static methods to required staticmethod decorator concept.

the static methods,dont takes any thing as a default parameter.

in static methods we are using some generalized data i.e., not related to class and instance.

we can access the static methods from out side the class directly by using classname or reference variable.

```
        syntax
        ------
        class classname:
                -------
                -------
                @staticmethod
                def methodname()
                        -------
                        -------
```

ex:

```
class student:
    col_name="Vagdevi"
    def __init__(self):
        self.sid=101
        self.sname="siva"
    @classmethod
    def col_info(cls):
        cls.col_add="Ameerpet"
        print(student.col_name)
        print(cls.col_add)
    def std_info(self):
        print(student.col_name)
        print(student.col_add)
        print(self.sid)
        print(self.sname)
    @staticmethod
    def add(x,y):
        print(x+y)
s1=student()
s1.col_info()
s1.std_info()
s1.add(4,5)
```

```
student.col_info()
student.add(10,20)

output:
------
Vagdevi
Ameerpet
Vagdevi
Ameerpet
101
siva
9
Vagdevi
Ameerpet
30

note:
-----
without staticmethod decorator,we can create a static methods directly.

ex:
---
class test:
    @staticmethod
    def m1():
        print("hai")
    def m2():
        print("hello")
t1=test()
test.m1()
t1.m1()
test.m2()
t1.m2()

output:
------
hai
hai
hello
Traceback (most recent call last):
  File "C:/Users/DELL/Desktop/demo.py", line 11, in <module>
    t1.m2()
TypeError: test.m2() takes 0 positional arguments but 1 was given
```