variables:
---
the variables are references,to store the data for future purpose or future use.

in object oriented mechanism,the variables can be categorized into 3-types,they are

1).Static/Class variables

2).Non-Static/Instance variables

3).Local/Method variables

Static/Class variables
---
we can define any one variable with-in the class and above the methods and constructor definition,that type of variables called Static/Class variables.

the static variables to allocate the memory space at only once.

which data is common for all the objects of a class,that data should be define as static variables.

we can access the static variables data with in the methods by using class name.

we can access the static variables data from outside the class by using reference variable or class name.

ex:

```
class student:
    col_name="Vagdevi"
    col_add="Hyderabad"
    def std_info(self):
        print(student.col_name)
        print(student.col_add)
s1=student()
s1.std_info()
print(student.col_name)
print(student.col_add)
print(s1.col_name)
print(s1.col_add)
```

output:

```
Vagdevi
Hyderabad
Vagdevi
Hyderabad
Vagdevi
Hyderabad
```

Non-Static/Instance variables:
---
we can define a variables with in the class and with in the method/constructor with the help of default parameter(self),that type of variables are called Non-Static Variables.

the non-static variables memory allocation is depends on the creation of no.of objects to that class.

```
        1 object        --> 1 time memory allocation

        2 object's      --> 2 time's memory allocation

        3 object's      --> 3 time's memory allocation

          ....                          .....

        N object's      --> N time's memory allocation
```

which data is changeing from one object to another object that type of data represented as Non-static variables.

we can access the non-static variables data with in the methods by using self(default parameter name).

we can access the non-static variables data from outside of that calss by using reference variable only.

ex:

```
class student:
    def std_info(self):
        self.sid=101
        self.sname="siva"
        print(self.sid)
        print(self.sname)
s1=student()
s1.std_info()
print(s1.sid)
print(s1.sname)
```

output:

```
101
siva
101
siva
```

Local/Method variables:

---
we can define a variables with in the methods directly,that type of variables are called Local Variables.

we can access the Local variables data with in that particular method only.

ex:

```
class test:
    x=10 #static
    def m1(self):
        self.y=20 #non-static
        z=30 #local
        print(test.x)
        print(self.y)
        print(z)
    def m2(self):
        print(test.x)
        print(self.y)
        print(z)
t1=test()
t1.m1()
t1.m2()
```

output:

```
10
20
30
10
20
Traceback (most recent call last):
  File "C:/Python310/e3.py", line 15, in <module>
    t1.m2()
  File "C:/Python310/e3.py", line 12, in m2
    print(z)
NameError: name 'z' is not defined
```

ex:

```
class student:
    col_name="Vagdevi"
    col_add="Hyderabad"
    def std_info(self):
        self.sid=101
        self.sname="siva"
        print(self.sid)
        print(self.sname)
        print(student.col_name)
        print(student.col_add)
```

```
s1=student()
s1.std_info()

output:
-------
101
siva
Vagdevi
Hyderabad

note:
----
whenever we are defineing the non-static varibales within the method in that case
the non-static variables memory allocation is depends on no.of times we are calling
that method with respect to one object.

in this case more-memory consumeing,to overcome that problem we are going to using
Constructor's concept.
```