what is Constructor?
----
        a constructor is a special kind of method,which is used to initialize the
non-static variables of class.

        in python,the constructor name should be __init__

        the constructor will be executed automatically at the time of creating an
object.

ex1:

```
class test:
    def __init__(self):
        print("i am in constructor")
    def m1(self):
        print("i am in m1")
t1=test()
```

output:

```
i am in constructor
```


        with respect to one object the constructor will be executed at only once
but with respect to one object we can call one method   N-no.of times.

ex2:

```
class test:
    def __init__(self):
        print("i am in constructor")
    def m1(self):
        print("i am in m1")
t1=test()
t1.m1()
t1.m1()
t1.m1()
t2=test()
t2.m1()
t2.m1()
```

output:

```
i am in constructor
i am in m1
i am in m1
i am in m1
i am in constructor
i am in m1
```

```
i am in m1

ex3:
---
class student:
    col_name="Vagdevi"
    col_add="Hyderabad"
    def __init__(self):
        self.sid=101
        self.sname="siva"
    def std_info(self):
        print(self.sid)
        print(self.sname)
        print(student.col_name)
        print(student.col_add)
s1=student()
s1.std_info()
s1.std_info()
s1.std_info()

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

ex4:
----
class student:
    col_name="Vagdevi"
    col_add="Hyderabad"
    def __init__(self,sid,name):
        self.sid=sid
        self.sname=name
    def std_info(self):
        print(self.sid)
        print(self.sname)
        print(student.col_name)
        print(student.col_add)
s1=student(101,'siva')
s1.std_info()
```

```
s2=student(102,'rama')
s2.std_info()

output:
------
101
siva
Vagdevi
Hyderabad
102
rama
Vagdevi
Hyderabad

ex5:
----
class cust:
    bname="Kotak"
    def __init__(self,acno,name):
        self.cacno=acno
        self.cname=name
        self.bal=0
    def deposit(self,amt):
        self.damt=amt
        self.bal+=self.damt
        print("hello %s,after deposit your current balance is\
 %.2f"%(self.cname,self.bal))
    def withdraw(self,amt):
        self.wamt=amt
        if self.wamt>self.bal:
            print("sorry %s,Insufficient Funds in Your Account,\
 please check your balance"%self.cname)
            x=input("do you continue to check the balance: ")
            if x=='yes':
                print("your current balance is:",self.bal)
            else:
                print("Thank you")
        else:
            self.bal-=self.wamt
            print("hello %s,after withdraw your current balance is\
 %.2f"%(self.cname,self.bal))
    def display(self):
        print("Bank Name:",cust.bname)
        print("Customer Name:",self.cname)
        print("Customer Acc.No:",self.cacno)
c1=cust(1001,'siva')
c1.deposit(3000)
c1.withdraw(2300)
c1.display()
print('*'*30)
```

```
c2=cust(1002,'rama')
c2.deposit(2000)
c2.withdraw(2300)
c2.display()
```

output1:

```
hello siva,after deposit your current balance is 3000.00
hello siva,after withdraw your current balance is 700.00
Bank Name: Kotak
Customer Name: siva
Customer Acc.No: 1001
****************************
hello rama,after deposit your current balance is 2000.00
sorry rama,Insufficient Funds in Your Account, please check your balance
do you continue to check the balance: yes
your current balance is: 2000
Bank Name: Kotak
Customer Name: rama
Customer Acc.No: 1002
```

output2:

```
hello siva,after deposit your current balance is 3000.00
hello siva,after withdraw your current balance is 700.00
Bank Name: Kotak
Customer Name: siva
Customer Acc.No: 1001
****************************
hello rama,after deposit your current balance is 2000.00
sorry rama,Insufficient Funds in Your Account, please check your balance
do you continue to check the balance: no
Thank you
Bank Name: Kotak
Customer Name: rama
Customer Acc.No: 1002
```

ex6:

```python
class cust:
    bname="Kotak"
    def __init__(self):
        self.cacno=int(input("enter Account Number: "))
        self.cname=input("enter Name: ")
        self.bal=0
    def deposit(self):
        self.damt=float(input("enter deposit ammount: "))
        self.bal+=self.damt
        print("hello %s,after deposit your current balance is\
 %.2f"%(self.cname,self.bal))
```

```python
    def withdraw(self):
        self.wamt=float(input("enter withdraw ammount: "))
        if self.wamt>self.bal:
            print("sorry %s,Insufficient Funds in Your Account,\
 please check your balance"%self.cname)
            x=input("do you continue to check the balance: ")
            if x=='yes':
                print("your current balance is:",self.bal)
            else:
                print("Thank you")
        else:
            self.bal-=self.wamt
            print("hello %s,after withdraw your current balance is\
 %.2f"%(self.cname,self.bal))
    def display(self):
        print("Bank Name:",cust.bname)
        print("Customer Name:",self.cname)
        print("Customer Acc.No:",self.cacno)
c1=cust()
c1.deposit()
c1.withdraw()
c1.display()
```

output:
------
enter Account Number: 1001
enter Name: siva
enter deposit ammount: 2000
hello siva,after deposit your current balance is 2000.00
enter withdraw ammount: 1200
hello siva,after withdraw your current balance is 800.00
Bank Name: Kotak
Customer Name: siva
Customer Acc.No: 1001

enter Account Number: 1002
enter Name: rama
enter deposit ammount: 3000
hello rama,after deposit your current balance is 3000.00
enter withdraw ammount: 3100
sorry rama,Insufficient Funds in Your Account, please check your balance
do you continue to check the balance: yes
your current balance is: 3000.0
Bank Name: Kotak
Customer Name: rama
Customer Acc.No: 1002

differencess between constructor's and method's:
----
                  constructor's

```
             -------------
```
1).the constructor is used to initialize the non-static variables.

2).the constructor name should be __init__

3).the constructor is executed automatically at the time of creating an object.

4).with respect to one object the constructor will be executed at only once.

```
             methods's
             ---------
```
1).the method is used to represent the Business logics to perform a particular operation.

2).we can take method name as any name or anything

3).the method is not executed automatically,whenever we are calling a method then only method logic will be executed.

4).with respect to one object we can call one method N-no.of time's