what is Garbage Collector?
---
the Garbage collector is a one predefined program,which is used to deallocate the memory space for unused or unreferenced objects automatically,this process is called Garbage collection.

once Garbage collector is executed more-memory space is available so that the rest of the program execution become Fast.

any high-level programming language which support Garbage collector concept,that type of programming languages are called Garbag Collected Langauges.

python inbuilt supporting Garbage collector concept.

in python,by default the garbage collector is enable.

ex1:
---
wap to check wheather the garbage collector is enable or not?

```
import gc
print(gc.isenabled())
```

output:

True

ex2:
---
wap to enable or disable the garbage collector manually?

```
import gc
print(gc.isenabled())
gc.disable()
print(gc.isenabled())
gc.enable()
print(gc.isenabled())
```

output:

True
False
True

when garbage collector is activated?
---
scenario-1:
---
once program execution is over then immediately our python memory manager internally to call's the garbage collector,the garbage collector to delete that

object's from the memory.

        before delete that objects from memory internally destructor is executed.

        in python, __del__ is a destructor.

ex1:

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

output:

```
C:\Users\DELL\Desktop>python demo.py
i am in constructor
bye
i am in destructor
```

note:
----
we can delete the objects manually by using 'del' keyword

ex2:

```
class test:
    def __init__(self):
        print("i am in constructor")
    def __del__(self):
        print("i am in destructor")
t1=test()
t2=test()
t3=test()
del t2
print("bye")
```

output:

```
C:\Users\DELL\Desktop>python demo.py
i am in constructor
i am in constructor
i am in constructor
i am in destructor
bye
i am in destructor
i am in destructor
```

scenario-2:
---
whenever object reference_count is Zero,then immediately our python memory manager to calls the garbage collector,the garbage collector to delete that object from the memory.

ex:

```
import sys
class test:
    def __init__(self):
        print("i am in constructor")
    def __del__(self):
        print("i am in destructor")
t1=test()
print(sys.getrefcount(t1))
t2=t1
print(sys.getrefcount(t1))
t3=t2
print(sys.getrefcount(t1))
t4=test()
print(sys.getrefcount(t4))
del t3
print(sys.getrefcount(t1))
del t2
print(sys.getrefcount(t1))
del t1
print("bye")
```

output:

```
C:\Users\DELL\Desktop>python demo.py
i am in constructor
2
3
4
i am in constructor
2
3
2
i am in destructor
bye
i am in destructor
```