Garbage Collection

What is garbage collection?

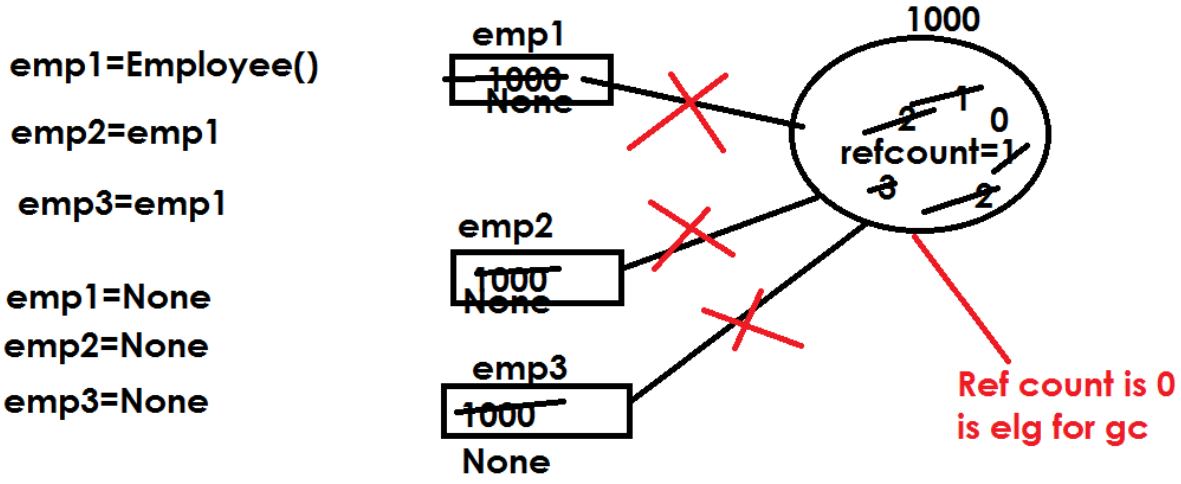
Garbage collection is process of de-allocating unused memory space or removing unused objects. In python memory management is automatic. It is taken care by python runtime. Python runtime provides one service called garbage collection.

Advantage

1. Reduce the chance of memory leak
2. Avoids wastage of memory

How garbage collection works in python?

In python garbage collection work using a method called reference count. If object bind with any reference variable, the reference count of that object incremented by one.



sys.getrefcount(object)

this method returns reference count of object (OR) object is bind with how many reference variables it called reference count

```
>>> import sys
>>> l1=[10,20,30]
>>> l2=l1
>>> sys.getrefcount(l1)
3
```

Destructor in python

__del__() method of object is class is called destructor

This method is executed automatically whenever object is garbage collected.
Destructor is used to de-allocate all the resources allocated within constructor
Constructor is executed whenever object is created
Destructor is executed before object is garbage collected

Example:

```
class Employee:
    def __init__(self):
        print("employee object is created...")
    def __del__(self):
        print("employee object is deleted")

def main():
    emp1=Employee()
    del emp1
    emp2=Employee()

main()
```

Output:

```
employee object is created...
employee object is deleted
employee object is created...
employee object is deleted
```

The gc module provides the following functions:

gc.enable()
Enable automatic garbage collection.

gc.disable()
Disable automatic garbage collection.

gc.isenabled()
Return True if automatic collection is enabled.

Example:

```
import gc
class Employee:
    def __init__(self):
        print("employee object is created...")
```

```
    def __del__(self):
        print("employee object is deleted")

def main():
    print(gc.isenabled())
    emp1=Employee()
    emp2=Employee()

gc.disable()
main()
input()
```

Output:

```
False
employee object is created...
employee object is created...
employee object is deleted
employee object is deleted
>>>
```