A Closure is a function Object that remembers values in enclosing scopes if they are not present in memory

```
# Inner Function
def outerFunction():
   print("Outer Function")

   def innerFunction():
      print("Inner Function")
   return innerFunction

result = outerFunction()
```

Output

```
Outer Function
```

Note:

In the example, we have outerFunction() function, which creates and returns a function

The nested innerFunction is a closure

Now here, result = outerFunction()

The outerFunction() returns a innerFunction() and assignes to the result variable.
At this moment it has finished its execution.
How ever, the innerFunction() closure still has access to the OuterFunction()

```
# Inner Function
def outerFunction():
   print("Outer Function")

   def innerFunction():
      print("Inner Function")
   return innerFunction

result = outerFunction()
```

```
Outer Function
```

```
print(result)
```

```
Output
<function outerFunction.<locals>.innerFunction at 0x0000008E1F9301F0>
```

Here, outerfunction locals the inner function at some address

```
print(result.__name__)
```

```
Output
innerFunction
```

__name__ is a built-in variable which evaluates to the name result variable

```
result()
```

```
Output
Inner Function
```

Now, here we are calling the result() function

Case 1:

```
def d1():
    x = 10
    print("d1 scope", x)
    def d2():
        y = 20
        print("d2 scope: ", x)
        print("d2 scope: ", y)
    return d2

z = d1()
```

Output

```
d1 scope 10
```

Case 2:

```
def d1():
    x = 10
    print("d1 scope", x)
    def d2():
        y = 20
        print("d2 scope: ", x)
        print("d2 scope: ", y)
    return d2

z = d1()
z()
```

Output

```
d1 scope 10
d2 scope:  10
d2 scope:  20
```

Case 3:

```
def d1():
    x = 10
    print("d1 scope", x)
    def d2():
        y = 20
        print("d2 scope: ", x)
        print("d2 scope: ", y)
    return d2

z = d1()
z()
del d1
d1()
```

Output

```
Traceback (most recent call last):
  File "E:\Github\Python Workspace\PythonWorkspace\Day12C_Closures\Eg3.py", line 13, in <module>
    d1()
NameError: name 'd1' is not defined
d1 scope 10
d2 scope:  10
d2 scope:  20
```

Case 4:

```
def d1():
    x = 10
    print("d1 scope", x)
    def d2():
        y = 20
        print("d2 scope: ", x)
        print("d2 scope: ", y)
    return d2

z = d1()
print("Before Delete")
z()
del d1
print("After Delete")
z()
```

Output

```
d1 scope 10

Before Delete

d2 scope:  10

d2 scope:  20

After Delete

d2 scope:  10

d2 scope:  20
```