**Experiment 9**

**AIM:**

Write a program to test this class. Declare a class cont with integer static member cnt to store the number of objects active, a constructor to increment it and a destructor to decrement it. Declare a static function show\_cnt() to display the value of cnt.

**Theory:**

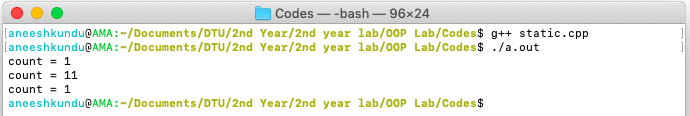
A static member is shared by all objects of the class. All static data is initialized to zero when the first object is created, if no other initialization is present. We can't put it in the class definition but it can be initialized outside the class by redeclaring the static variable, using the scope resolution operator :: to identify which class it belongs to.

By declaring a function member as static, you make it independent of any particular object of the class. A static member function can be called even if no objects of the class exist and the static functions are accessed using only the class name and the scope resolution operator ::.

**Code:**

1. #include < iostream >
2. using namespace std;
3. class cont {
4. static int cnt;
5. public:
6. cont() {
7. cnt++;
8. }
9. ~cont() {
10. cnt--;
11. }
12. static int get\_count() {
13. return cnt;
14. }
15. };
16. int cont::cnt;
17. int main() {
18. cont a;
19. cout << "count = " << cont::get\_count() << endl;
20. cont \* b = new cont[10];
21. cout << "count = " << cont::get\_count() << endl;
22. delete[] b;
23. cout << "count = " << cont::get\_count() << endl;
24. return 0;
25. }

**Output:**

****

**Discussion:**

The variable cnt is a static data member, and the function get\_count() is a static function that returns number of active objects and can be accessed only through the class and not its instances.