

SDF II(15B11CI211)

EVEN Semester 2021



2nd Semester , First Year

Jaypee Institute Of Information Technology (JIIT), Noida

Lecture 6 – Destructor and static member and function

Destructor

- Destructor is also a member function of the class.
- It work opposite to constructor.
- It is call to destroy the object.
- It has a same as the class name but having a tild sign before the name.
- It also does not have any return type.
- It does not take argument,
- It should declare in the public section of the class.

Example

```
#include <iostream>

using namespace std;

class Employee {
    public:
        Employee()
        { cout<<"Constructor Invoked"<<endl; }

        ~Employee()
        { cout<<"Destructor Invoked"<<endl; } };

    int main(void)
    {
        Employee e1; //creating an object of Employee

        Employee e2; //creating an object of Employee

        return 0; }
```

<https://www.geeksforgeeks.org/destructors-c/>

Output

Constructor Invoked

Constructor Invoked

Destructor Invoked

Destructor Invoked

Point to remember

- There can not be more than one destructor in the program unlike constructor.
- Destructor call automatically.
- We do not create a destructor untill we have define memory by dynamic memory allocation or pointer. For that we define destructor to destroy object and release memory.

Static Variable

- Static variable start with static keyword.
- Static variable declared once in a program and its scope remains end of the program.
- Even if the function is called multiple times, space for the static variable is allocated only once and the value of variable in the previous call gets carried through the next function call

// Static variable in a function

```
#include <iostream>

using namespace std;

void test()
{
    static int count = 0; // static variable
    cout << count ;      // value is updated and // will be carried to next // function calls
    count++; }

int main()
{
    for (int i=0; i<5; i++)
        test();
    return 0; }
```


Static variable in a class

- Static variable inside a class should be initialized explicitly by using the class name and scope resolution operator outside the class.

Example

```
#include <iostream>

using namespace std;

class Test
{
public:
    static int i;

    Test() { // Do nothing    }; };

int Test::i = 1;

int main()
{
    Test obj;

    cout << obj.i; }


```

Output

1

Static function in a class

Static member functions are allowed to access only the static data members or other static member functions they can not access the non-static data members or member functions of the class.

Example

```
#include <iostream>

using namespace std;

class Test
{ public:
    static void show()
    { cout<<"This is test program for static functions!";  });

int main()
{ // invoking a static member function
Test::show();
}
```

Output

This is test program for static functions!

References

- S. Arora, Computer science in C++. 2002