# C++ Destructors

**C++ destructor** is a special member function that is executed automatically when an object is destroyed that has been created by the [constructor](http://www.trytoprogram.com/cplusplus-programming/constructors/). C++ destructors are used to de-allocate the memory that has been allocated for the object by the constructor.

## **Destructor in c++ programming**

Its syntax is same as constructor except the fact that it is preceded by the tilde sign.

~class\_name() { }; //syntax of destructor

## **Structure of C++ destructors**

/\*...syntax of destructor....\*/

class class\_name

{

public:

class\_name(); //constructor.

~class\_name(); //destructor.

}

Unlike constructor a destructor neither takes any arguments nor does it returns value. And destructor can’t be overloaded.

## **C++ Destructor Example**

/\*.....A program to highlight the concept of destructor.......... \*/

#include <iostream>

using namespace std;

class ABC

{

public:

ABC () //constructor defined

{

cout << "Hey look I am in constructor" << endl;

}

~ABC() //destructor defined

{

cout << "Hey look I am in destructor" << endl;

}

};

int main()

{

ABC cc1; //constructor is called

cout << "function main is terminating...." << endl;

/\*....object cc1 goes out of scope ,now destructor is being called...\*/

return 0;

} //end of program

**Output**

Hey look I am in constructor

function main is terminating....

Hey look I am in destructor

**Explanation**

In the above program, when constructor is called “Hey look I am in constructor” is printed then following it “Function main is terminating…..” is printed but after that the object cc1 that was created before goes out of scope and to de-allocate the memory consumed by cc1 destructor is called and “Hey I am in destructor” is printed.

Note: Remember that more than one destructor can’t be used in a program. Only single destructor is allowed.