Destructor

Private Destructor

Virtual Destructor

Pure Virtual Destructor

Destructor vs member functions

Multiple Destructors in one class ?

When to write user defined Destructor ?

Can a Destructor be virtual ?

**-------------------------------------------------------------------------**

**Destructor:-**

In C++, a **destructor** is a special member function that is automatically called when an object goes out of scope or is explicitly deleted. Its primary purpose is to release resources that the object may have acquired during its lifetime (such as memory, file handles, or network connections). Here are key points about destructors in C++:

**Key Characteristics:**

1. **Naming Convention**: The destructor's name is the same as the class's name, but it is prefixed with a tilde (~). For example, if the class is MyClass, the destructor would be named ~MyClass().
2. **No Return Type**: Destructors do not have a return type, not even void.
3. **No Parameters**: Destructors cannot have parameters, meaning you can't overload destructors.
4. **Automatic Invocation**: Destructors are automatically invoked when an object:
   * **Goes out of scope (for stack-allocated objects).**
   * **Is explicitly deleted (for heap-allocated objects using delete).**
5. **Default Destructor**: If a class does not define a destructor, the compiler generates a default one, which may suffice if the class doesn't manage resources directly (such as dynamic memory allocation).
6. **Destructor for Base and Derived Classes**: In inheritance hierarchies, destructors for derived classes are called first, followed by destructors of base classes. If you're working with inheritance and base class pointers, it's crucial to make base class destructors virtual to ensure proper cleanup.

The thing is to be noted here, if the object is created by using new or the constructor uses new to allocate memory that resides in the heap memory or the free store, the destructor should use delete to free the memory.

Destructors release memory locations and resources when the program ends.

Destructor is used to release resources and is automatically created by the compiler if not defined manually.

#include <iostream>

class MyClass {

public:

MyClass() {

std::cout << "Constructor called!" << std::endl;

}

~MyClass() {

std::cout << "Destructor called!" << std::endl;

}

};

int main() {

MyClass obj; // Constructor is called here

// Destructor will be called automatically when obj goes out of scope

return 0;

}

When is a destructor called?

A destructor function is called automatically when:

the object goes out of scope

the program ends

a scope (the { } parenthesis) containing local variable ends.

a delete operator is called

Destructor rules

The name should begin with a tilde sign(~) and match the class name.

There cannot be more than one destructor in a class.

Unlike constructors that can have parameters, destructors do not allow any parameter.

They do not have any return type, not even void. I

A destructor should be declared in the public section of the class.

The programmer cannot access the address of the destructor.

It has no return type, not even void.

When you do not specify any destructor in a class, the compiler generates a default destructor and inserts it into your code.

**When to Define a Destructor:**

You should define a custom destructor when your class manages resources that need to be released manually (e.g., dynamically allocated memory or file streams). If no custom destructor is provided, and the class uses only built-in types or objects that automatically manage their resources (e.g., smart pointers), a destructor might not be needed.

**Constructor & Destructor Call Order**

Constructors are called in order, but destructors are called in reverse order.

9 
lø 
11 
12 
13 
14 
15 
16 
17 
18 
19 
2ø 
21 
22 
23 
24 
25 
26 
27 
public: 
Customer (string name) 
this—>name = name; 
cout<<"constructor 
int main() 
Customer Al(' 
is 
is 
2 
3 
rohitnegiaohits4a 
++ cd "/Users/roh 
top/ C++/' • g++ 
Class3 "/Users/ 
esktop/C++/"CIass3 
constructor is 1 
constructor is 2 
constructor is 3 
Destructor is 3 
Destructor is 2 
Destructor is Ir 
rohitnegi80hitß-Ma 
O 

29 
3ø 
31 
32 
33 
int 
main ( ) 
Customer 
Customer 
new Customer; 
1 
Destructor is 3 
Destructor is 2 
Destructor is 1 
rohitnegi80hits4 
• ++ cd "/Users/r 
top/C++/" g++ 
Class3 "/User: 
esktop/C++/"Ctass: 
constructor is 1 
constructor is 2 
constructor is 3 
Destructor is 3 
Destructor is 2 
Destructor is 1 

Note:A4 ka destructor call nhi hua h

Th usko resolve krene ke liye "delete"

26 
27 
28 
29 
30 
31 
32 
33 
34 
35 
int 
main ( ) 
Customer Al( 
Customer 
delete A4; 
new Customer; 
rohitnegi80hits4a 
++ cd "/Users/roh 
top/C++/•• g++ C I 
Class3 "/Users,' 
esktop/C++/"Class3 
constructor is I 
constructor is 2 
constructor is 3 
constructor is 4 
Destructor is 4 
Destructor is 3 
is 2 
Destructor is I 
rohi t neg imohits-".a 