

Constructor vs Destructor

Constructor is a special kind of class member function that is executed when an object of that class is instantiated.

Destructor is another special kind of class member function that is executed when an object of that class is destroyed. They are the counterpart to constructors. When a variable goes out of scope, or a dynamically allocated variable is explicitly deleted using delete operator, the class destructor is called (if it exists) to help clean up the class before it is removed from ~~the~~ memory.

For simple classes, a destructor is not needed because C++ will automatically cleanup the memory for us. However, if we have dynamically allocated memory, or if ~~we~~ we need to do some kind of maintenance before the class is destroyed, the destructor is the perfect place to do so.

Like constructors, destructors have specific naming rules:—

- (1) The destructor must have the same name as the class, preceded by a tilde (~).
- (2) The destructor cannot take arguments.
- (3) The destructor has no return type.

In this way, destructor can exist only one in a class, and hence it cannot be overloaded. like constructors. Since constructors have many different types of parameters according to need of the object instantiation.