

Destructor

Destructor usually deletes any extra resources allocated by the [object](#).

Note, in Symbian C++:

- Destructor can NOT be allowed to leave! A leave from a destructor may cause the program to terminate abruptly. In S60 3rd edition and later releases a leave is an exception and there can be only one active exception at a time. The destructor may have been called as a part of a routine to handle an exception (leave) and raising another exception will cause the program to be terminated.
- Always delete objects your class owns, from the class destructor.
- Don't delete objects that you don't own (those that you merely use).
- Don't allocate twice (this will cause a memory leak).
- Don't delete twice (this will corrupt the heap).
- When you delete outside of the destructor, immediately set the pointer to zero.
- When you are reallocating, you must use the sequence 'delete, set pointer to zero, allocate', just in case the allocation fails.

See also [delete](#).