C class

A **C** class is one derived from class CBase.

The CBase class has the following main characteristics:

- **Data members** are always **zeroed** upon construction if an object is created on the heap because operator new() is overloaded.
- CBase has a **virtual destructor**
- CBase has private copy constructor and assignment operator

Thus, the following characteristics of C classes arise:

- They should always be allocated on the heap to ensure zero-initialization.
- They inherit the virtual destructor of the base class, so can be guaranteed to be cleaned up correctly.
- They need an explicit copy constructor or assignment operator if there is clear intention that a particular class will support copying

C classes typically have ownership of other C classes or resources that need cleanup. Since these resources often have non-trivial construction and, because of the possibility that a leave might occur during the construction process, C classes typically use the two-phase construction idiom.

- It is good practice that any class that derives from other non 'C' classes lists its base classes in the correct order. CBase or a CBase-derived class should be the first base class in the list:

```
class CCustomClass : public CBase, public MClass1
```

or, for a class further down the inheritance tree:

```
class CAnotherClass : public CCustomClass, public MClass2
```

This will emphasize the primary inheritance tree. If any other class is the first base class, you may have problems when using the cleanup stack.

See Also

- Fundamentals of Symbian C++/Class Types & Declarations#C Classes
- An Introduction to C Classes

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.