

General Steps in Converting C application to C++

John Tal - Feb 12, 1992

1. Create Class Definition

1.1 Identify Private And Protected Members

Identify all variables which are passed through functions but are really private to the class. These include:

- o Work Areas
- o Head Pointers
- o Counters

Place these in the private or protected class sections.

1.2 Install Public Access To Private Members

Identify all private or protected class member variables which interact with the application. All reading or writing of non-public members should be accomplished by a public class member function. There should be NO access to non-public members by the application.

1.3 Identify Non-Public Functions

All functions which are not directly called by the application should be private or protected.

1.4 Modify Class Function Prototypes

Remove from all functions parameters which are now embodied by class member data. This includes the same items as in 1.1.

1.5 Create Class Constructor And Destructor

These can be VOID and NULL functions are whatever is required by the class. All allocated memory should be freed in the destructor.

1.6 Determine Class Access Within Hierarchies

There will be elementary classes which you will use to build more sophisticated classes through different methods. Each method will influence the security and general object access to be defined. It can also be helpful to distinguish between what a class contains and what it does in refining class interfaces.

1.6.1 Simple Inclusion

In this case a class is constructed with other classes as members.

1.6.2 Derived Classes

This includes the areas of multiple-inheritance and polymorphism. The derived class will inherit the class attributes of its parent class(es).

1.6.3 Differentiated Access

You may discover a situation where you want the application to be unable to access certain class areas but you want another class to be able to access all class areas. One solution is to use a friend modifier (NOT RECOMMENDED, the friend modifier breaks many object-oriented rules). The best solution may be to create a server class which will then access others classes for the application.

2 Modify C Source Code

2.1 Rename Or Copy Code

Rename or copy the c source files from *.c to whichever extension is used by your C++ compiler (usually *.cpp).

2.2 Modify Code Comments

Comments in the source code can be left in the /* style, but for those who work in both C and C++, having native C++ comments in // style makes it easier to switch between languages.

2.3 Modify Functions

2.3.1 Modify Function Interface

Modify function interfaces to no longer pass items such as in 1.1.

2.3.2 Modify Function Comment Header Block

Modify comment header block to include security of class member (private, protected, public). Carefully examine any function who's description includes the words 'and' or 'or'. These are signs of cohesion problems. (If a function calculates data AND writes it to disk, the function should be broken up into two functions, one to calculate and one to write.)

2.3.3 Break Up Functions

Create new functions from those displaying cohesion or coupling problems which can be corrected.

2.3.4 Create Interface Functions For Class Private Data

Create functions necessary for an application to read or write (get/set) data which is in private or public class security segments.