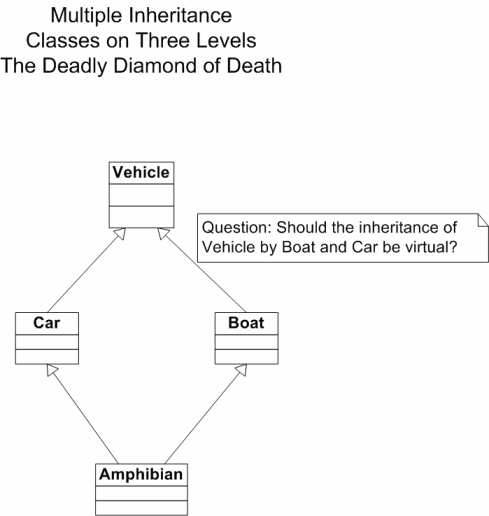
**What is the Deadly Diamond of Death?**

In the simple case of multiple inheritance shown above, there is usually no problem. But what if the two base classes themselves inherit from a single base class? In this case, the diagram, which is shown below, is sometimes called the "Deadly Diamond of Death" (or, by the more charitable, the "Deadly Diamond of Derivation") for reasons that we will now discuss. Begin by looking at the following diagram:



The problem with the situation shown in the above diagram is that since every data member in the Vehicle class is inherited by both the Car and the Boat classes, and the Amphibian class contains everything in a Car as well as everything in a Boat, there will be two of every data member in Vehicle in Amphibian, leading to ambiguity and confusion. To compound the difficulty, it may sometimes make sense to have these two copies and sometimes not.

For example, even an abstract Vehicle may be regarded as having a maximum speed, and an amphibious vehicle may have a different maximum speed as a car (land vehicle) than it has as a boat (water vehicle). So in this case it actually makes sense to have two "maximum speed" data members in the Amphibian class.

On the other hand, a Vehicle might also have a data member recording the year of manufacture, and we would want only one of these in our Amphibian class.

This is still a pretty simple example, but it is already getting messy, since we have at least three cases:

1. We want more than one copy of each data member from the top level class.
2. We want to ensure that we have only one copy of each data member from the top level class.
3. We want more than one copy of some data members from the top level class, but only one copy of the others.

http://cs.stmarys.ca/~porter/csc/common\_341\_342/notes/oop\_multiple\_inheritance.html