Multiple Inheritance

In this lesson, we'll learn about multiple inheritance.

WE'LL COVER THE FOLLOWING

- General rules of multiple inheritance
- Diamond-problem
- Virtual base class

Multiple inheritance is a feature of some object-oriented computer programming languages in which an object or class can inherit characteristics and features from more than one parent object or class. It is different from single inheritance, where an object or class may only inherit from one particular object or class.

General rules of multiple inheritance

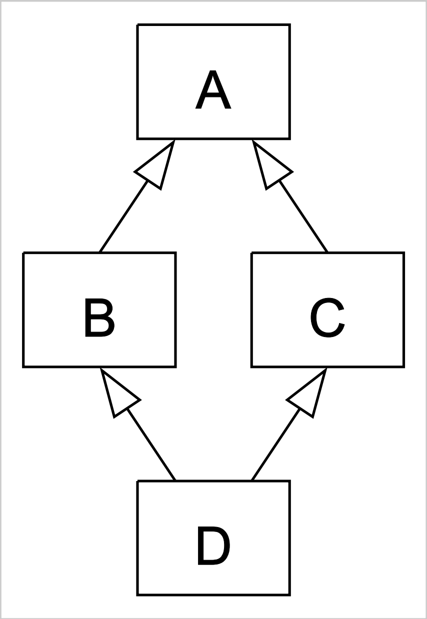
- By providing a list of base classes will result in multiple inheritance.
- Multiple inheritance is a generalization of single inheritance.
- The access rights for each base class can be specified individually.
- For classes, access rights are `private` by default; for structs, access rights are public by `default`.

If a derived class has more than one instance of a base class, the invocation of its members is ambiguous.

Diamond-problem

The **diamond problem** is an ambiguity that arises when two classes `B` and `C` inherit from `A`, and class `D` inherits from both `B` and `C`. If there is a method in `A` that `B` and `C` have overridden, and `D` does not override it, then which version of the method does `D` inherit: that of `B`, or that of `C`?

It is called the “diamond problem” because of the shape of the class inheritance diagram in this situation. Class `A` is at the top, both `B` and `C` are separately beneath it, and `D` joins the two together at the bottom to form a diamond shape.



Diamond Problem

> Ambiguous calls to members can be resolved by using the scope operator.

Virtual base class

A virtual base class solves the problem of multiple inheritance because the derived class gets only one base class. When a class is derived virtually from a base class it becomes a virtual base class.

In the next lesson, we’ll look at some examples of multiple inheritance.