**Item32: Make sure public inheritance models “is-a.”**

If you write that class D (“Derived”) publicly inherits from class B (“Base”), you are telling C++

that every object of type D is also an object of type B, but *not vice versa*. You are asserting that anywhere an object of type B can be used, an object of type D can be used just as well, because every object of type D *is* an object of type B. On the other hand, if you need an object of type D, an object of type B will not do: every D is-a B, but not vice versa.

class Person { ... };

class Student: public Person { ... };

We know from everyday experience that every student is a person, but not every person is a student.