Copy Constructor Encyclopedia Article

Copy Constructor

Contents

Copy Constructor

A **constructor** is a special **function** belonging to a **class** that builds objects of that class from nothing. A "copy constructor" is a special kind of constructor that makes an **object** that is an identical copy of another. It builds the second object by copying the state of the first object into the second.

The copy constructor usually takes as its sole input **parameter** a reference to an object of its own type. In other words, a copy constructor for the lunchBox class would take a reference to an already-existing lunchBox as its input and create an identical copy of it.

In **programming** parlance this is sometimes referred to as the "X X ref" constructor, where "ref" is short for "reference," where the "reference" is a special kind of **identifier** that "refers" to an object and allows the program to **access** the object, but is not the object itself. In **C++** the reference is denoted by the ampersand symbol, &, and the declaration for a copy constructor for the lunchBox class might look like this:

- lunchBox(const lunchBox& theBox);

Programmers often fail to realize how important the copy constructor is, and, moreover, how important it is to **code** it correctly. For example, a C++ object that contains a char*, or a pointer to a **string**, cannot safely copy the **value** of the pointer because there would then be two objects with apparently different **data** members that are actually the same. When one or other of the objects changes the string, or worse, destroys it, the other object will then contain unexpectedly changed data. The best the programmer can hope for in these circumstances is a catastrophic program failure; other consequences might be much worse.

For this reason it is good programming practice to write a copy constructor for every class in the program and allow contained objects to copy themselves rather than have the containing object assume this responsibility. In the above example, the char* would be replaced by a properly implemented String class that would "know" how to make a copy of another String object.