The name \texttt{Retort()}, on the other hand, is an abbreviation of the qualified name, and it can be used only in certain circumstances, such as in the code for the class methods.

Another way of describing this situation is to say the name \texttt{Retort} has class scope, so the scope resolution operator is needed to qualify the name when it is used outside the class declaration and a class method.

To create an object, which is a particular example of a class, use the class name as if it were a type name:

\begin{verbatim}
Bozo bozetta;
\end{verbatim}

This works because a class \textit{is} a user-defined type.

A class member function, or method, is invoked by a class object. You do so by using the dot membership operator:

\begin{verbatim}
cout << Bozetta.Retort();
\end{verbatim}

This invokes the \texttt{Retort()} member function, and whenever the code for that function refers to a particular data member, the function uses the value that member has in the \texttt{bozetta} object.

\section*{Class Constructors and Destructors}

Meanwhile, there's more to be done with the \texttt{Stock} class. There are certain standard functions, called \textit{constructors} and \textit{destructors}, that you should normally provide for a class. Let's see why they are needed and how to write them.

One of C++'s aims is to make using class objects similar to using standard types. However, you can't yet initialize a \texttt{Stock} object the way you can an ordinary \texttt{int} or \texttt{struct}:

\begin{verbatim}
int year = 2001;             // okay
struct thing
{
  char * pn;
  int m;
};
\end{verbatim}