# A Lisp-Like Linked List Class[*]

Jeffrey D. Oldham

2000 Jan 15

Andrew Koenig presents a Lisp-like linked list C++ class in chapter 15 of *Ruminations on C++*. This is a short introduction to using the class (header file and implementation file).

## 1 The Linked List Class

Unlike some programming languages, the C++ programming language requires explicit type declarations. Thus, in the following, I will refer to lists of `int`s, but lists of any other type can also be created.

A list is

- either an empty list
- or an item followed by a list.

### 1.1 Empty Lists

To create an empty list, use

```
Seq<int>()
```

To check if a list `L` is empty, use

```
L.empty()
```

which returns the boolean value true if the list has no items and otherwise false. Using `L` in a place where a boolean is expected yields true if the list is not empty and otherwise false. For example, `if (L) cout << "list is not empty\n";`.

### 1.2 Nonempty Lists

To add an integer, e.g., 3, to the an existing list `L`, use

```
Seq<int>(3,L)
```

To check if a list is nonempty, negate the result of checking for an empty list.

To obtain a list `L`'s first item, which has `int` type, use

```
L.hd()
```

To obtain the rest of the list, which has `Seq<int>` type, use

```
L.tl()
```

[*]

### 1.3 Example

The `subst` function substitutes one string for another string in a linked list of `string`s.

```
#include "seq.h"        // Note the "", not <>, causes this
                        // directory to be searched as well as the
                        // standard places implied by <>.
#include <string>

// Given a list of strings, return a (new) list with one string
// substituted for the other string.
Seq<string> subst(const string & oldString, const string & newString,
                  const Seq<string> & SL) {
  if (SL.empty())
    return SL;
  else if (!SL.empty()) {
    if (SL.hd() == oldString)
      return Seq<string>(newString, subst(oldString, newString, SL.tl()));
    else
      return Seq<string>(SL.hd(), subst(oldString, newString, SL.tl()));
  }
}
```

## 2 Logistics of Using the Code

To use the linked list class with a program you wrote, copy the two files (header file and implementation file) to the directory containing the program's C++ code. One way to do this is to use the "Save As..." item on a WWW browser's file menu.

Another way is to issue the shell command `wget http://www.cs.trinity.edu/˜joldham/1321/lectures/lists/seq.h http://www.cs.trinity.edu/˜joldham/1321/lectures/lists/seq.cc`. The wget program copies the specified WWW links to your local directory. See also the wget manual. Isn't wget slick?

In your C++ program, add the line

```
#include "seq.h"
```

near the other header inclusions. See also this sample program.