

## CS173: Intermediate Programming

### List ADT

Overview	
<p>The List class is an ordered collection of items. The items in the list must all be of the same datatype which is specified by a template.</p> <p>The number of items in the list is specified by the length() of the list. Items in the list are indexed from 0 (first item) to length-1 (last item). The list grows dynamically so that it is never full.</p> <p>Items may be inserted into the list and removed from the list. Values at individual indices may be accessed and/or changed.</p> <p>If a method is attempted with an invalid index (removing, adding, indexing), an error message is printed and the program terminates.</p>	
Constructors	
default	<p>A new empty List is created.</p> <pre>List l;</pre>
copy	<p>Create a new List from an existing one.</p> <pre>List l1(l2);</pre>
Access	
isEmpty	<p>Returns true if the list is empty, false otherwise.</p> <pre>if (l.isEmpty()) ...</pre>
length	<p>Returns the number of items in the list.</p> <pre>for ( int i = 0; i &lt; l.length(); i++) ...</pre>
operator[]	<p>Accesses (by reference) the item at the specified index.</p> <pre>int i = l[5]; l[3] = 10;</pre> <p>A program-termination error is issued if the index is invalid. Invalid indices are less than 0 or greater than length-1.</p>
cout <<	<p>Overload the cout &lt;&lt; operator to print the list.</p> <pre>cout &lt;&lt; l1 &lt;&lt; endl;</pre> <p>Formating example:    [ 1, 2, 3 ]</p>

Modifiers	
append	Appends a new item onto the back of the list. <code>l1.append(5);</code>
insert(item,pos)	Inserts a new value at the specified position. Valid indices for the position are 0 ... length. Inserting at position=length is like an append. Existing values in the list are moved up one index location to make room for the new item. <code>l1.insert(5,1); // inserts 5 at index 1</code> A run-time error is generated for an invalid index.
remove	Removes an item at the specified location. Valid locations are 0...length-1; <code>l1.remove(3);</code> A run-time error is generated for an invalid index.
operator+	Concatenates two lists into a new list. Does not change either existing list. <code>l3 = l1 + l2;</code>
operator=	Assignment operator. <code>l1 = l2;</code>
clear	Removes all items from the list. <code>l1.clear();</code>
destructor	Cleans up the memory of the list.