

C++ Library Reference Sheet

<p>Lexicon</p> <pre>Lexicon lex; Lexicon english(filename); lex.addWord(word); bool present = lex.contains(word); bool pref = lex.containsPrefix(prefix); int numElems = lex.size(); bool empty = lex.isEmpty(); lex.clear();</pre>	<p>Map</p> <pre>Map<K, V> map = {{k₁, v₁}, ... {k_n, v_n}}; map[key] = value; // Autoinsert bool present = map.containsKey(key); int numKeys = map.size(); bool empty = map.isEmpty(); map.remove(key); map.clear(); Vector<K> keys = map.keys();</pre>
<p>Stack</p> <pre>stack.push(elem); T val = stack.pop(); T val = stack.peek(); int numElems = stack.size(); bool empty = stack.isEmpty(); stack.clear();</pre>	<p>Queue</p> <pre>queue.enqueue(elem); T val = queue.dequeue(); T val = queue.peek(); int numElems = queue.size(); bool empty = queue.isEmpty(); queue.clear();</pre>
<p>Set</p> <pre>Set<T> set = {v₁, v₂, ..., v_n}; set.add(elem); set += elem; set -= elem; Set<T> result = set - elem; // or + elem bool present = set.contains(elem); set.remove(x); set -= x; set -= set2; Set<T> unionSet = s1 + s2; Set<T> intersectSet = s1 * s2; Set<T> difference = s1 - s2; T elem = set.first(); int numElems = set.size(); bool empty = set.isEmpty(); set.clear();</pre>	<p>Vector</p> <pre>Vector<T> vec = {v₁, v₂, ..., v_n}; vec.add(elem); vec += elem; vec.insert(index, elem); vec.indexOf(elem); // index or -1 vec.remove(index); vec.clear(); vec[index]; // Read/write int numElems = vec.size(); bool empty = vec.isEmpty(); vec.subList(start, numElems);</pre>
<p>string</p> <pre>str[index]; // Read/write str.substr(start); str.substr(start, numChars); str.find(c); // index or string::npos str.find(c, startIndex); str += ch; str += otherStr; str.erase(index, length);</pre>	<p>ifstream</p> <pre>input.open(filename); input >> val; getline(input, line);</pre> <p>GWindow</p> <pre>GWindow window(width, height); gw.drawLine(x0, y0, x1, y1); pt = gw.drawPolarLine(x, y, r, theta);</pre>
<p>GPoint</p> <pre>double x = pt.getX(); double y = pt.getY();</pre>	<p>Point</p> <pre>int x = pt.getX(); int y = pt.getY();</pre>
<p>TokenScanner</p> <pre>TokenScanner scanner(source); while (scanner.hasMoreTokens()) { string token = scanner.nextToken(); ... } scanner.ignoreWhitespace();</pre>	<p>General Utility Functions</p> <pre>int getInteger(<i>optional-prompt</i>); double getReal(<i>optional-prompt</i>); string getLine(<i>optional-prompt</i>); int randomInteger(lowInclusive, highInclusive); double randomReal(lowInclusive, highExclusive); error(message); x = max(val1, val2); y = min(val1, val2); stringToInteger(str); stringToReal(str); to_string(intVal); to_string(realVal);</pre>