```
// $Id: lexicalcast.cpp,v 1.6 2015-02-05 19:08:00-08 - - $

//
// Illustrate to_string, from_string, lexical_cast.
// Assume the use of << and >> on sstreams.
//

#include <iostream>
#include <sstream>
#include <stdexcept>
#include <string>
#include <typeinfo>

using namespace std;

//
// convert a thing into a string.
//

template <typename item_t>
string to_string (item_t that) {
   ostringstream stream;
   stream << that;
   return stream.str ();
}

//
// Scan a string to grab a thing.
//

template <typename item_t>
item_t from_string (const string &that) {
   stringstream stream;
   stream << that;
   item_t result;
   bool converted = stream >> result  // Is string is a valid item_t?
                 && stream >> std::ws // Flush trailing white space.
                 && stream.eof();     // Must now be at end of stream.
   if (not converted) {
      throw domain_error (string (typeid (item_t).name())
            + " from_string (" + that + ")");
   }
   return result;
}

//
// Lexically cast a thing to a string then to another thing.
//

template <typename target_t, typename source_t>
target_t lexical_cast (source_t that) {
   return from_string (to_string (that));
}

//
// Main.
//

```

```
int main () {
   cout << to_string<double> (9) << endl;
   cout << from_string<int> ("42") << endl;
   return 0;
}

//TEST// lexicalcast >lexicalcast.out 2>&1
//TEST// mkpspdf lexicalcast.ps lexicalcast.cpp lexicalcast.out

```

```
9
42
```