```
#include "keyitem.h"

KeyItem::KeyItem( string name, unsigned int value, unsigned int numKeys )
: Item( name, value), numKeys( numKeys )
{
    this->keyGroup  = new bool[numKeys];
    this->setAllKeys( false );
}

KeyItem::~KeyItem()
{
    cerr << "~KeyItem";
    delete [] keyGroup;
    keyGroup    = NULL;
}

bool KeyItem::setKey( unsigned int index, bool value )
{
    if( index < this->numKeys )
    {
        this->keyGroup[index]   = value;
        return true;
    }

    return false;
}

void KeyItem::setAllKeys( bool value )
{
    for( unsigned int uintI; uintI < this->numKeys; uintI++ )
    {
        this->keyGroup[uintI]   = value;
    }
}

bool KeyItem::getKey( unsigned int index ) throw( string )
{
    if( index < this->numKeys )
    {
        return this->keyGroup[index];
    }
    else
    {
        throw string( "INVALID KEY INDEX" );
    }
}

unsigned int KeyItem::getNumKeys()
{
    return this->numKeys;
```

```
51  }
52
53  /*  toString, displays name, value, and values of all keys in keyGroup
54                (uses Item::toString in its implementation)
55                  format -- NAME, $VALUE, [VALUE VALUE VALUE ...]
56                  EXAMPLE -- room keys, 10, [yes no yes]
57                  */
58  string KeyItem::toString()
59  {
60      stringstream ss;
61      ss << this->getName() << ", $" << this->getValue() << ", [";
62      for( unsigned int uintI = 0; uintI < this->getNumKeys(); uintI++ )
63      {
64          if( this->getKey( uintI ) )
65          {
66              ss << "yes";
67          }
68          else
69          {
70              ss << "no";
71          }
72
73          if( (uintI + 1) < this->getNumKeys() )
74              ss << " ";
75      }
76
77      ss << "]";
78
79      return ss.str();
80  }
81
```