```
// Implementation for Library

#include "library.h"

//Default constructor. There is no book initially
Library::Library()
{
        Book * p = first;
        first = NULL;
        numOfBooks = 0;
}

//Copy constructor
//Clone all the books in the existing library rhs and make a new library
Library::Library(const Library & rhs)
{
        Book * p = rhs.first;
        p = NULL;
        numOfBooks = rhs.numOfBooks;
        //Clone all the books in the existing library rhs and make a new library
        //Using insertBook(Book * b) might be easy. For example,

        while (p != NULL)
        {
                Book * bb = new Book(p->getISBN());
                insertBook(bb);
                p = p->getNext();
//              p->setNext(p);

                numOfBooks++;
        }
}

//Delete all books and make this library empty
Library::~Library()
{
        //You already have makeEmpty() function. Does it help here?
        makeEmpty();
}

void Library::printLibrary()
{
  Book * p = first;

  while(p != NULL)
  {
          cout << p->getISBN() << endl;
          p = p->getNext();
```

```
  }
}


//Delete all books and make this library empty
void Library::makeEmpty()
{
        //You already have deleteBook(int s) function. Does it help here?
        /*while (first != NULL)
        {
                for (int i = 0; i < numOfBooks; i++)
                {
                        deleteBook(i);
                        first->getNext();
                }
        }*/
        Book * tempPtr;

        while (first != NULL)
        {
                tempPtr = first;
                first = first->getNext();
                delete tempPtr;
        }
        numOfBooks = 0;
}

void Library::insertBook(Book * b)
{
        b->setNext(first);
        first=b;
        numOfBooks++;
}

void Library::deleteBook(int s)
{
        Book* location = first;
        Book* tempLocation;

        // Locate node to be deleted.
        if (s == first->getISBN())
        {
                tempLocation = location;      // Save pointer to node
                first = first->getNext();    // Delete first node.
        }
        else
        {
```

```
                while (s != location->getNext()->getISBN())
                        location = location->getNext();

                // Delete node at location->next
                tempLocation = location->getNext();
                location->setNext(tempLocation);
        }
        delete tempLocation;     // Return node
        numOfBooks--;
}

int Library::numBooksInLibrary()
{
        return numOfBooks;
}

bool Library::searchBook(int s)
{
        Book * p = first;
        for (int i = 0; i < numOfBooks; i++)
        {
                if (s == p->getISBN())
                        return true;
                else
                        p = p->getNext();
        }
        return false;
}
```