CSE 331/ ONSAY

HOMEWORK 6

100 points

DUE DATE: November 3rd 11:59pm.

Warning: For any homework assignment that contains a programming segment please read the following very carefully.

Your code must compile and run on Black server. If your code does not work on Black server as submitted the grade for that problem is 0. Always test your code on Black, even if it is incomplete, make sure to get your code to compile and run on our Servers.

This homework contains 1 problem.

With your requirements document you are given:

1. An incomplete Library.h
2. main.cpp
3. Makefile

Please do not modify main unless it is specified in the problem

Problem 1

Your assignment is to build a book management system for a library. This library has books. Each book has an ISBN, which is a unique 13digit number, a title, and author's name, and the number of copies of the book available. You need to support the following abilities. When you are implementing these methods listed below make sure to supply documentation for each method with pre and post conditions (see guidance additional guidance for pre post documentation at the end of this requirements document for a reference, you can also use the Queue lecture notes for reference):

- Search for a book
- Add a book to the library
- Check in a book
- Check out a book.
- Get number of available copies of a book

The books will be stored in a binary search tree to support efficient searching and insertion. This library never intentionally removes a book from their collection, so deleting a book is not a required operation.

When a book is added to the library, its number of available copies is initialized to 1. When a book is checked out, decrement the number of available copies and return a pointer to the book. Of course, if there are no available copies of the book, you cannot check out the book, so return a nullptr.
When checking in a book, the number of available books is incremented by 1.
If the book does not exist in the library, you cannot check it in.
Do not create a new BookNode in the tree. We do not want to steal people's books that they accidently give us!

You will use the skeleton project provided to implement this design. Do not modify signatures of any public methods, but feel free to add custom private methods as you see necessary.

Remember all codes written in .h files must compile and run with our main method on Black Server.

Homework 6 Deliverables:

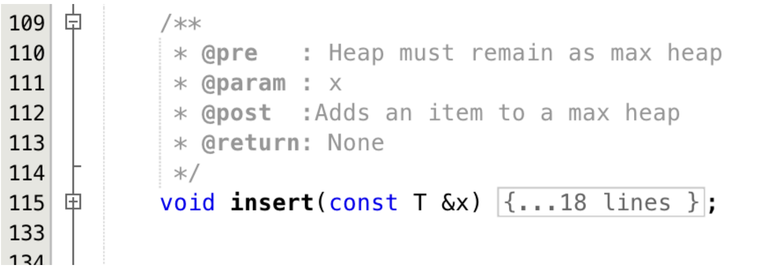
The following files must be submitted via Handin no later than November 3rd 2016 by 11:59pm

1. **Library.h**

DOCUMENTING YOUR CODE WITH PRE AND POST CONDITIONS

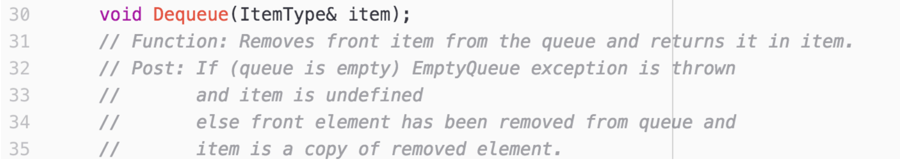
BOTH DOCUMENTATION STYLES ARE ACCEPTABLE.PICK ONE!

EXAMPLE 1 Documentation is done as a header to a function.



```
/**
 * @pre    : Heap must remain as max heap
 * @param : x
 * @post   :Adds an item to a max heap
 * @return: None
 */
void insert(const T &x) {...18 lines };
```

EXAMPLE 2: Documentation is done during listing of public methods in a class.



```
void Dequeue(ItemType& item);
// Function: Removes front item from the queue and returns it in item.
// Post: If (queue is empty) EmptyQueue exception is thrown
//       and item is undefined
//       else front element has been removed from queue and
//       item is a copy of removed element.
```