CS 201 DATA STRUCTURES
ASSIGNMENT 3
SECTION C&F
Fall 2022

DUE: Monday Nov 21, 2022

NOTE: Late submissions will not be accepted. Please do submit your code on Google Classroom, no email submission will be accepted.

TO SUBMIT: Documented and well written code in C++. Undocumented code will be assigned a zero.

In this Assignment, you will simulate some functionality of a library management system to efficiently process the records that are books. A book record contains the following information:

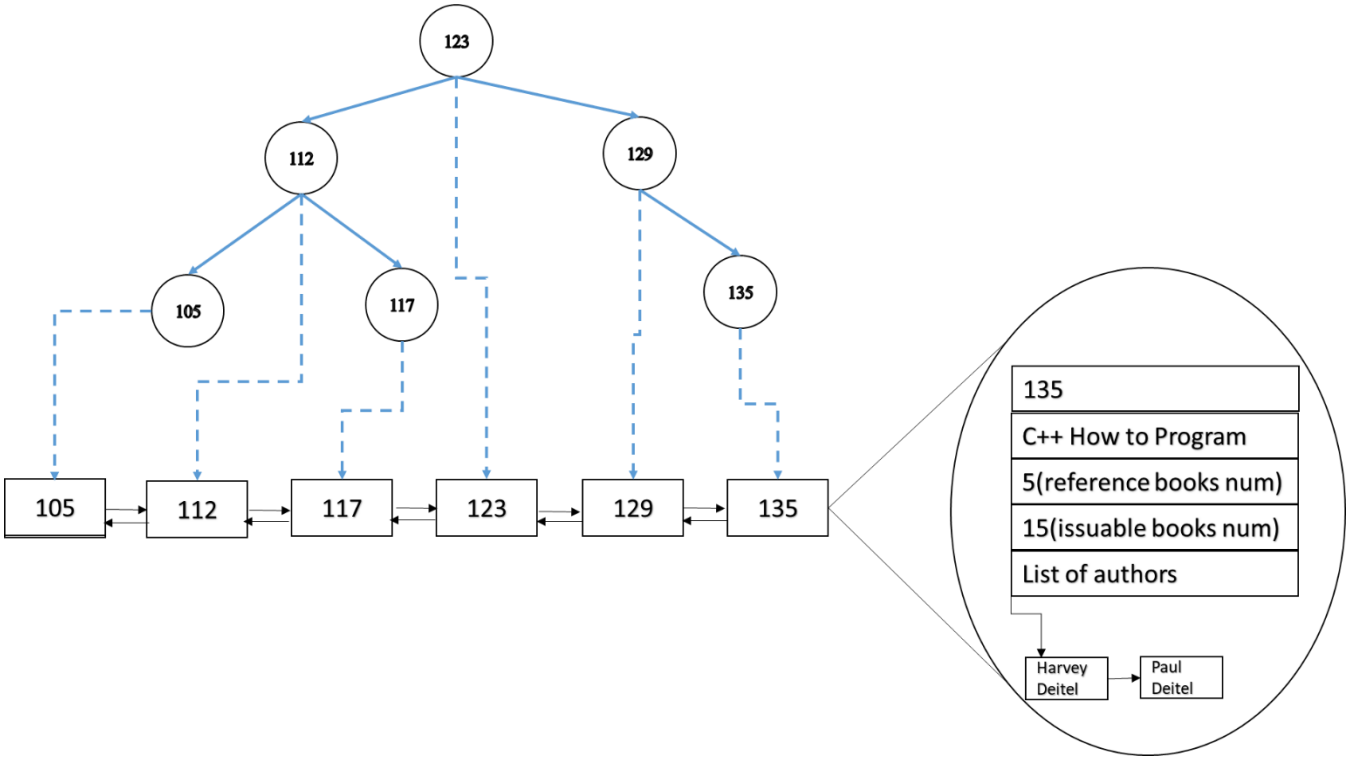
a) ISSN (a unique number assigned to books)
b) Title
c) List of Author name(s)
d) #of copies available as reference book
e) # of copies available as issuable book

A number of users want to access the library online and search for the books. In order to efficiently perform the different operations, an index is created. The index is created using AVL tree data structure, where each node of the tree contains ISSN and the location of the book record in the main memory. The records of all books are stored in a doubly linked-list sorted according to the ISSN.

The following functions can be performed on the database:

1. Create a book record (C) - a new book record is added to the tree as well as the in the doubly linked list in the sorted order.
2. Review a book record (R) -- Given an ISSN display the complete book record
3. Update a book record (U) -- Given ISSN, any field in the book record can be updated except the ISSN
4. Delete a book record (D) -- Given an ISSN delete the book record from the tree and from the list.

Make sure that all your operations are performed in O(lgn) worst case time.



The figure given above is an illustrative example of how data is stored and organized. All the records of the books are stored in a sorted doubly linked list. An index is created on the ISSN using an AVL tree that keeps the ISSN as well as the address of the node of the doubly linked list that keeps the entire record. The dotted links are pointers to the nodes of the doubly linked list. The oval in the bottom right is the magnified view of a node in the doubly linked list.