**Assignment No 4**

**Title:**

A book consists of chapters, chapters consist of sections and sections consist of subsections. Construct a tree and print the nodes. Find the time and space requirements of your method.

**Objectives:**

1. To understand concept of tree data structure

2. To understand concept & features of object oriented programming.

**Learning Objectives:**

* To understand concept of class
* To understand concept & features of object oriented programming.
* To understand concept of tree data structure.

**Learning Outcome:**

* Define class for structures using Object Oriented features.
* Analyze tree data structure.

**Software Required:** g++ / gcc compiler- / 64 bit Fedora, eclipse IDE

**Input:** Book name & its number of sections and subsections along with name.

**Output: Formation of tree structure for book and its sections.**

**Theory:**

**Definition:**

A tree T is a set of nodes storing elements such that the nodes have a parent-child relationship that satisfies the following

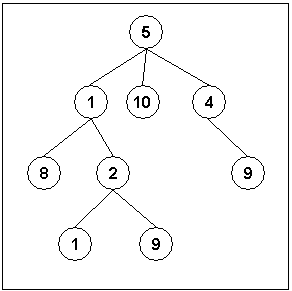
• if T is not empty, T has a special tree called the root that has no parent

• each node v of T different than the root has a unique parent node w; each node with parent w is a child of w

Tree is a widely-used data structure that emulates a tree structure with a set of linked nodes. The tree graphicaly is represented most commonly as on *Picture 1*. The circles are the nodes and the edges are the links between them.

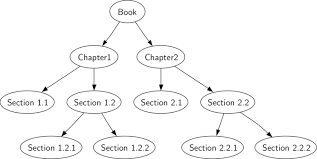
Trees are usualy used to store and represent data in some hierarhical order. The data are stored in the nodes, from which the tree is consisted of.The topmost node in a tree is called the root node.

A subtree is a portion of a tree data structure that can be viewed as a complete tree in itself.



*Fig1. An example of a tree*

 For this assignment we are considering the tree as follows.



Structure of every node in this tree is:

struct book\_node

{

char title[15];

int chapt\_count;

struct book\_node \*down[10];

}

**Algorithm:**

**Creating a Tree:**

1)Allocate memory for new book node.

2)store following details in the node

book name and no of chapters

3)for total no of chapters do

i)Allocate memory for new chapter node and assign its address as downpointer of book node.

ii)store chapter name and total no of sections in it.

iii) for total no of sections do

1. Allocate memory for new section node and assign its address as downpointer of chapter node.
2. store section name and total no of subsections in it.

c)for total number of subsections do

-Allocate memory for new sub section node and assign its address as down pointer of section node.

-Store details of subsections in the node.

**Display Tree:**

1)If root is not NULL

i)Display Book Title

ii)for total no of chapters do

a)Display chapter name

b) for total no of sections do

- Display section name

- for total number of subsections do

-Store name of subsections in the node.

**Advantages of trees**

Trees are so useful and frequently used, because they have some very serious advantages:

* Trees reflect structural relationships in the data
* Trees are used to represent hierarchies
* Trees provide an efficient insertion and searching
* Trees are very flexible data, allowing to move subtrees around with minumum effort

**Conclusion:** This program gives us the knowledge tree data structure.