Assignment 3

```
#include <iostream>
#include <string.h>
using namespace std;

struct node
{
   string label;
   int ch_count;
   struct node *child[10];
} * root;

class GT
{
public:
   void create_tree();
   void display(node *r1);

   GT()
   {
      root = NULL;
   }
};

void GT::create_tree()
{
   int tbooks, tchapters, i, j, k;
   root = new node;
   cout << "Enter name of book : ";
   cin.get();
   getline(cin, root->label);
   cout << "Enter number of chapters in book : ";
   cin >> tchapters;
   root->ch_count = tchapters;
   for (i = 0; i < tchapters; i++)
   {
      root->child[i] = new node;
      cout << "Enter the name of Chapter :" << i + 1 << " : ";
      cin.get();
      getline(cin, root->child[i]->label);
      cout << "Enter number of sections in Chapter : " << root->child[i]->label << " : ";
      cin >> root->child[i]->ch_count;
      for (j = 0; j < root->child[i]->ch_count; j++)
      {
         root->child[i]->child[j] = new node;
         cout << "Enter Name of Section :" << j + 1 << " : ";
         cin.get();
         getline(cin, root->child[i]->child[j]->label);
      }
   }
}
```

```
void GT::display(node *r1)
{
   int i, j, k, tchapters;
   if (r1 != NULL)
   {
      cout << "\n-----Book Hierarchy---";
      cout << "\n Book title : " << r1->label;
      tchapters = r1->ch_count;
      for (i = 0; i < tchapters; i++)
      {

         cout << "\nChapter " << i + 1;
         cout << " : " << r1->child[i]->label;
         cout << "\nSections : ";
         for (j = 0; j < r1->child[i]->ch_count; j++)
         {
            cout << "\n"<< r1->child[i]->child[j]->label;
         }
      }
   }
   cout << endl;
}

int main()
{
   int choice;
   GT gt;
   while (1)
   {
      cout << "-----------------" << endl;
      cout << "Book Tree Creation" << endl;
      cout << "-----------------" << endl;
      cout << "1.Create" << endl;
      cout << "2.Display" << endl;
      cout << "3.Quit" << endl;
      cout << "Enter your choice : ";
      cin >> choice;
      switch (choice)
      {
      case 1:
         gt.create_tree();
      case 2:
         gt.display(root);
         break;
      case 3:
         cout << "Thanks for using this program!!!";
         break;
      default:
         cout << "Wrong choice!!!" << endl;
      }
   }
   return 0;
```

}

```
#include <iostream>
#include <string.h>
using namespace std;

struct node
{
    string label;
    int ch_count;
    struct node *child[10];
} * root;

class GT
{
public:
    void create_tree();
    void display(node *r1);

    GT()
    {
        root = NULL;
    }
};

void GT::create_tree()
{
    int tbooks, tchapters, i, j, k;
    root = new node;
    cout << "Enter name of book : ";
    cin.get();
    getline(cin, root->label);
    cout << "Enter number of chapters in book : ";
    cin >> tchapters;
    root->ch_count = tchapters;
    for (i = 0; i < tchapters; i++)
    {
        root->child[i] = new node;
        cout << "Enter the name of Chapter :" << i
```

```
student@student-OptiPlex-3010:~/Desktop$ cd Nikita
student@student-OptiPlex-3010:~/Desktop/Nikita$ g++ Ass3.cpp
student@student-OptiPlex-3010:~/Desktop/Nikita$ ./a.out
-----------------
Book Tree Creation
-----------------
1.Create
2.Display
3.Quit
Enter your choice : 1
Enter name of book : DSA
Enter number of chapters in book : 2
Enter the name of Chapter :1 : Binary Tree
Enter number of sections in Chapter : Binary Tree : 2
Enter Name of Section :1 : Threaded Binary Search
Enter Name of Section :2 : Binary Tree Search
Enter the name of Chapter :2 : Graphs
Enter number of sections in Chapter : raphs : 2
Enter Name of Section :1 : Directional Graph
Enter Name of Section :2 : Multigraph

-----Book Hierarchy---
 Book title : DSA
Chapter 1 : Binary Tree
Sections :
Threaded Binary Search
inary Tree Search
Chapter 2 : raphs
Sections :
Directional Graph
ultigraph
-----------------
Book Tree Creation
-----------------
1.Create
2.Display
3.Quit
Enter your choice : 3
Thanks for using this program!!!-----------------
Book Tree Creation
-----------------
```