

/******

Subject : DSA Laboratory

Class: SE Computer Engineering

Practical No: 03 - A C++ Program to implement General Tree

Construct a Tree of Book consist of Chapters,
Chapters consist of Sections
and Sections consist of Subsections.

*****/

```
        //....Header Files
#include <iostream>
using namespace std;

        //....Node Structure
struct Node
{
    char name[25];
    struct Node *link1;
    struct Node *link2;
    struct Node *link3;
}*Root, *Ch;

        //....Function to Create General Tree
void create_Tree()
{
    struct Node *temp;

    temp = new struct Node; //...Allocate memory

    cout<<"\n\n Enter Book Name: "; //....Insert Data and initialise Pointers
    cin>>temp->name;
    temp->link1 = NULL;
    temp->link2 = NULL;
    temp->link3 = NULL;

    if(Root == NULL) //.....Attach New Node in Tree
        Root = temp;
}

        //....Function to Display General Tree
void display()
{
    if(Root)
        cout<<"\n\n Book Name: "<<Root->name;

    Ch = Root->link1; //....Chapter:01
    if(Ch)
    {
        cout<<"\n\n Chapter 01: "<<Ch->name;
        if(Ch->link1)
```

```

    cout<<"\n\t Section 01: "<<Ch->link1->name;
    if(Ch->link2)
    cout<<"\n\t Section 02: "<<Ch->link2->name;
    if(Ch->link3)
    cout<<"\n\t Section 03: "<<Ch->link3->name;
}

```

```

Ch = Root->link2; //....Chapter:02
if(Ch)
{
    cout<<"\n\n Chapter 02: "<<Ch->name;
    if(Ch->link1)
    cout<<"\n\t Section 01: "<<Ch->link1->name;
    if(Ch->link2)
    cout<<"\n\t Section 02: "<<Ch->link2->name;
    if(Ch->link3)
    cout<<"\n\t Section 03: "<<Ch->link3->name;
}

```

```

Ch = Root->link3; //....Chapter:03
if(Ch)
{
    cout<<"\n\n Chapter 03: "<<Ch->name;
    if(Ch->link1)
    cout<<"\n\t Section 01: "<<Ch->link1->name;
    if(Ch->link2)
    cout<<"\n\t Section 02: "<<Ch->link2->name;
    if(Ch->link3)
    cout<<"\n\t Section 03: "<<Ch->link3->name;
}
}

```

```

//....Function to Insert Chapters in Book.
void insert_Chapter()
{
    struct Node *temp;

    temp = new struct Node; //...Allocate memory

    cout<<"\n\t Enter Chapter Name: "; //....Insert Data and initialise Pointers
    cin>>temp->name;
    temp->link1 = NULL;
    temp->link2 = NULL;
    temp->link3 = NULL;

    if(Root->link1 == NULL) //.....Attach New Node in Tree
        Root->link1 = temp;
    else if(Root->link2 == NULL)
        Root->link2 = temp;
    else if(Root->link3 == NULL)
        Root->link3 = temp;
}

```

```

else
    cout<<"\n ...No More Chapters can be added...!!!";
}

//....Function to Insert Sections in Chapters of Book.
void insert_Sec()
{
    struct Node *temp;

    temp = new struct Node; //...Allocate memory

    cout<<"\n\t Enter Section Name: "; //....Insert Data and initialise Pointers
    cin>>temp->name;
    temp->link1 = NULL;
    temp->link2 = NULL;
    temp->link3 = NULL;

    if(Ch->link1 == NULL) //.....Attach New Node in Tree
        Ch->link1 = temp;
    else if(Ch->link2 == NULL)
        Ch->link2 = temp;
    else if(Ch->link3 == NULL)
        Ch->link3 = temp;
    else
        cout<<"\n ...No More Sections can be added...!!!";
}

//....Function to Insert Sections in Chapters of Book.
void insert_Section()
{
    int chap;

    cout<<"\n\n Enter Sections of Chapter(1/2/3): ";
    cin>>chap;

    switch(chap)
    {
        case 1: Ch = Root->link1;
                insert_Sec();
                insert_Sec();
                break;
        case 2: Ch = Root->link2;
                insert_Sec();
                insert_Sec();
                break;
        case 3: Ch = Root->link3;
                insert_Sec();
                insert_Sec();
                break;
    }
}

```

```

}

//....Main Function
int main()
{
    cout<<"\n ----- A C++ Program to implement General Tree-----";

    Root = NULL;

    create_Tree();

    insert_Chapter();
    insert_Chapter();
    insert_Chapter();

    insert_Section();
    insert_Section();
    insert_Section();

    display();

    return 0;
}

/*
----- A C++ Program to implement General Tree-----

```

Enter Book Name: Data_Structures

Enter Chapter Name: Ch01_Array

Enter Chapter Name: Ch02_Stack

Enter Chapter Name: Ch03_Queue

Enter Sections of Chapter(1/2/3): 1

Enter Section Name: 1D_Array

Enter Section Name: 2D_Array

Enter Sections of Chapter(1/2/3): 2

Enter Section Name: Sequectial_Stack

Enter Section Name: Liked_Stack

Enter Sections of Chapter(1/2/3): 3

Enter Section Name: Simple_Queue

Enter Section Name: Circular_Queue

Book Name: Data_Structures

Chapter 01: Ch01_Array

Section 01: 1D_Array

Section 02: 2D_Array

Chapter 02: Ch02_Stack

Section 01: Sequectial_Stack

Section 02: Liked_Stack

Chapter 03: Ch03_Queue

Section 01: Simple_Queue

Section 02: Circular_Queue

...Program finished with exit code 0

Press ENTER to exit console.

*/