

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  struct node // Node Declaration
6  {
7      string label;
8      int ch_count; // Number of child nodes (chapters, sections, or subsections)
9      struct node *child[10]; // Child nodes (can be chapters, sections, or
subsections)
10 } *root;
11
12 class GT // Class Declaration
13 {
14 public:
15     void create_tree();
16     void display(node *r1);
17
18     GT()
19     {
20         root = NULL;
21     }
22 };
23
24 void GT::create_tree()
25 {
26     int tchapters, tsections, tsubsections, i, j, k;
27
28     root = new node; // Create the root node
29     cout << "Enter name of book: ";
30     cin.get();
31     getline(cin, root->label);
32
33     cout << "Enter number of chapters in book: ";
34     cin >> tchapters;
35     root->ch_count = tchapters;
36
37     for (i = 0; i < tchapters; i++)
38     {
39         root->child[i] = new node; // Create a new node for each chapter
40         cout << "Enter the name of Chapter " << i + 1 << ": ";
41         cin.get();
42         getline(cin, root->child[i]->label);
43
44         cout << "Enter number of sections in Chapter " << root->child[i]->label <<
": ";
45         cin >> tsections;
46         root->child[i]->ch_count = tsections;
47
48         for (j = 0; j < tsections; j++)
49         {
50             root->child[i]->child[j] = new node; // Create a new node for each
section
51             cout << "Enter Name of Section " << j + 1 << " in Chapter " << root-
>child[i]->label << ": ";
52             cin.get();
53             getline(cin, root->child[i]->child[j]->label);
54

```

```

55         cout << "Enter number of subsections in Section " << root->child[i]-
>child[j]->label << ": ";
56         cin >> tsubsections;
57         root->child[i]->child[j]->ch_count = tsubsections;
58
59         for (k = 0; k < tsubsections; k++)
60         {
61             root->child[i]->child[j]->child[k] = new node; // Create a new node
for each subsection
62             cout << "Enter Name of Subsection " << k + 1 << " in Section " <<
root->child[i]->child[j]->label << ": ";
63             cin.get();
64             getline(cin, root->child[i]->child[j]->child[k]->label);
65         }
66     }
67 }
68 }
69
70 void GT::display(node *r1)
71 {
72     int i, j, k, tchapters, tsections, tsubsections;
73
74     if (r1 != NULL)
75     {
76         cout << "\n----- Book Hierarchy -----";
77         cout << "\nBook title: " << r1->label;
78         tchapters = r1->ch_count;
79
80         for (i = 0; i < tchapters; i++)
81         {
82             cout << "\nChapter " << i + 1 << ": " << r1->child[i]->label;
83             tsections = r1->child[i]->ch_count;
84
85             for (j = 0; j < tsections; j++)
86             {
87                 cout << "\n Section " << j + 1 << ": " << r1->child[i]->child[j]-
>label;
88                 tsubsections = r1->child[i]->child[j]->ch_count;
89
90                 for (k = 0; k < tsubsections; k++)
91                 {
92                     cout << "\n Subsection " << k + 1 << ": " << r1->child[i]-
>child[j]->child[k]->label;
93                 }
94             }
95         }
96     }
97     cout << endl;
98 }
99
100 int main()
101 {
102     int choice;
103     GT gt;
104
105     while (true)
106     {
107         cout << "-----" << endl;
108         cout << "Book Tree Creation" << endl;
109         cout << "-----" << endl;
110         cout << "1. Create" << endl;

```

```
111     cout << "2. Display" << endl;
112     cout << "3. Quit" << endl;
113     cout << "Enter your choice: ";
114     cin >> choice;
115
116     switch (choice)
117     {
118     case 1:
119         gt.create_tree();
120         break;
121     case 2:
122         gt.display(root);
123         break;
124     case 3:
125         cout << "Thanks for using this program!" << endl;
126         return 0; // Exit the program
127     default:
128         cout << "Wrong choice!" << endl;
129     }
130 }
131 return 0;
132 }
133
```