**ASSIGNMENT NO 3  
TITLE:**  Implementation of general tree

CODE:

#include <iostream>

#include <string>

using namespace std;

struct node {

string label;

int ch\_count;

node\* child[10];

} \*root;

class GT {

public:

void create\_tree();

void display(node\* r1);

GT() {

root = NULL;

}

};

void GT::create\_tree() {

int tchapters, tsections, tsubsections;

root = new node;

cout << "Enter name of book: ";

cin.ignore();

getline(cin, root->label);

cout << "Enter number of chapters in book: ";

cin >> tchapters;

root->ch\_count = tchapters;

for (int i = 0; i < tchapters; i++) {

root->child[i] = new node;

cout << "Enter name of Chapter " << i + 1 << ": ";

cin.ignore();

getline(cin, root->child[i]->label);

cout << "Enter number of sections in Chapter \"" << root->child[i]->label << "\": ";

cin >> tsections;

root->child[i]->ch\_count = tsections;

for (int j = 0; j < tsections; j++) {

root->child[i]->child[j] = new node;

cout << "Enter name of Section " << j + 1 << ": ";

cin.ignore();

getline(cin, root->child[i]->child[j]->label);

cout << "Enter number of subsections in Section \"" << root->child[i]->child[j]->label << "\": ";

cin >> tsubsections;

root->child[i]->child[j]->ch\_count = tsubsections;

for (int k = 0; k < tsubsections; k++) {

root->child[i]->child[j]->child[k] = new node;

cout << "Enter name of Subsection " << k + 1 << ": ";

cin.ignore();

getline(cin, root->child[i]->child[j]->child[k]->label);

root->child[i]->child[j]->child[k]->ch\_count = 0;

}

}

}

}

void GT::display(node\* r1) {

if (r1 != NULL) {

cout << "\n----- Book Hierarchy -----";

cout << "\nBook Title: " << r1->label;

for (int i = 0; i < r1->ch\_count; i++) {

node\* chapter = r1->child[i];

cout << "\n Chapter " << i + 1 << ": " << chapter->label;

for (int j = 0; j < chapter->ch\_count; j++) {

node\* section = chapter->child[j];

cout << "\n Section " << j + 1 << ": " << section->label;

for (int k = 0; k < section->ch\_count; k++) {

node\* subsection = section->child[k];

cout << "\n Subsection " << k + 1 << ": " << subsection->label;

}

}

}

cout << endl;

}

}

int main() {

int choice;

GT gt;

while (true) {

cout << "\n---------------------------\n";

cout << "Book Tree Management System\n";

cout << "---------------------------\n";

cout << "1. Create Book Tree\n";

cout << "2. Display Book Tree\n";

cout << "3. Quit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

gt.create\_tree();

break;

case 2:

gt.display(root);

break;

case 3:

cout << "Thanks for using this program!" << endl;

return 0;

default:

cout << "Invalid choice. Try again.\n";

}

}

return 0;

}  
  
OUTPUT:



