# CourseList.cpp

#include "CourseNode.h"  
#include <iostream>  
#include <sstream>  
  
class CourseList {  
private:  
 CourseNode\* head;  
  
 int extractCourseNumber(const std::string& courseCode) {  
 std::stringstream ss;  
 for (char ch : courseCode) {  
 if (isdigit(ch)) ss << ch;  
 }  
 int number;  
 ss >> number;  
 return number;  
 }  
  
public:  
 CourseList() : head(nullptr) {}  
  
 ~CourseList() {  
 deleteAllNodes();  
 }  
  
 void addCourse(const std::string& courseCode, int creditHours, char grade) {  
 CourseNode\* newNode = new CourseNode(courseCode, creditHours, grade);  
 int newCourseNumber = extractCourseNumber(courseCode);  
  
 if (!head || extractCourseNumber(head->courseCode) > newCourseNumber) {  
 newNode->next = head;  
 head = newNode;  
 return;  
 }  
  
 CourseNode\* current = head;  
 while (current->next && extractCourseNumber(current->next->courseCode) < newCourseNumber) {  
 current = current->next;  
 }  
  
 newNode->next = current->next;  
 current->next = newNode;  
 }  
  
 void deleteCourse(const std::string& courseCode) {  
 if (!head) return;  
  
 if (head->courseCode == courseCode) {  
 CourseNode\* temp = head;  
 head = head->next;  
 delete temp;  
 return;  
 }  
  
 CourseNode\* current = head;  
 while (current->next && current->next->courseCode != courseCode) {  
 current = current->next;  
 }  
  
 if (current->next) {  
 CourseNode\* temp = current->next;  
 current->next = temp->next;  
 delete temp;  
 }  
 }  
  
 void deleteAllNodes() {  
 while (head) {  
 CourseNode\* temp = head;  
 head = head->next;  
 delete temp;  
 }  
 }  
  
 void displayList() const {  
 CourseNode\* current = head;  
 while (current) {  
 std::cout << "Course: " << current->courseCode  
 << ", Credit Hours: " << current->creditHours  
 << ", Grade: " << current->grade << "\n";  
 current = current->next;  
 }  
 if (!head) std::cout << "The list is empty.\n";  
 }  
};