According to the given question, we need to load the given file into the appropriate data structure and use it to print the course information as specified.

In the given file, at last add the -1 to determine end of file. good() and !eof() are also used to determine end of file but there is a possibility that extra line is read resulting in undesired output. We have to read file line by line so we will use getline() function for it. We get a string containing the current line. As information is separated by commas(',') so we need to split it into list. Just insert the characters till there is a delimiter into a list. Skip the delimiter and repeat this process until the end of the string is reached.

After getting all the required information insert the course object into list of courses.

For searching the course by course name, we can use linear search and one found then print the course information. For sorting the list by course number use the bubble sort.

**Note:** Make sure that abcu.txt and cpp file are in same folder.

**Contents of abcu.txt:**

MATH201,Discrete Mathematics  
CSCI300,Introduction to Algorithms,CSCI200,MATH201  
CSCI350,Operating Systems,CSCI300  
CSCI101,Introduction to Programming in C++,CSCI100  
CSCI100,Introduction to Computer Science  
CSCI301,Advanced Programming in C++,CSCI101  
CSCI400,Large Software Development,CSCI301,CSCI350  
CSCI200,Data Structures,CSCI101  
-1

**Code:**

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

// Definition of structure course

struct Course{

string courseNumber;

string name;

vector<string> prerequisites;

};

// Function to split string to list of strings on the basis of given delimiter

vector<string> tokenize(string s, string del = " ")

{

vector<string> stringArray;

int start = 0;

int end = s.find(del);

while (end != -1) {

stringArray.push\_back(s.substr(start, end - start));

start = end + del.size();

end = s.find(del, start);

}

stringArray.push\_back(s.substr(start, end - start));

return stringArray;

}

// Function to load file and store the details into list of courses

vector<Course> LoadDataStructure()

{

// Creating an object of iftsream class to open file

ifstream fin("abcu.txt",ios::in);

vector<Course> courses;

string line;

// Infinite loop

while(1)

{

getline(fin,line);

// if end of file is reached then break the loop

if(line=="-1")

break;

Course course;

// getting tokenized information which is separated by commas

vector<string> tokenizedInformation=tokenize(line,",");

// Storing information on the structure course

course.courseNumber=tokenizedInformation[0];

course.name =tokenizedInformation[1];

// if there are prerequisites then storing them too

for(int i=2;i<tokenizedInformation.size();i++)

{

course.prerequisites.push\_back(tokenizedInformation[i]);

}

// appending the course into list of courses

courses.push\_back(course);

}

// closing the file

fin.close();

// return the list of courses

return courses;

}

// printing course information of given course in proper format

void printCourse(Course course)

{

string courseNumber= course.courseNumber;

string name=course.name;

vector<string> prerequisites=course.prerequisites;

cout<<"Course Number: "<<courseNumber<<endl;

cout<<"Course Name: "<<name<<endl;

cout<<"Prerequisites: ";

for(int i=0;i<prerequisites.size();i++)

{

cout<<prerequisites[i]<<" ";

}

cout<<"\n\n";

}

// printing course information of all courses in proper format

void printCourseList(vector<Course> courses)

{

int n=courses.size();

// Using bubble sort to sort the list

for(int i=0;i<n-1;i++)

{

for(int j=0;j<n-i-1;j++)

{

if(courses[j].courseNumber > courses[j+1].courseNumber)

{

swap(courses[j+1],courses[j]);

}

}

}

// traversing list of courses to print all courses information

for(int i=0;i<n;i++)

{

printCourse(courses[i]);

}

}

// search the course for the user entered course number

void searchCourse(vector<Course> courses)

{

int n=courses.size();

string courseNumber;

int f=0;

cout<<"Enter courseNumber: ";

cin>>courseNumber;

for(int i=0;i<n;i++)

{

// if course found then print it

if(courses[i].courseNumber==courseNumber)

{

f=1;

printCourse(courses[i]);

break;

}

}

// if course with given course name not found then print error message

if(f==0)

{

cout<<"Course with given course number not found\n";

}

}

int main(int argc, char \*\*argv)

{

vector<Course> courses;

// Printing menu

cout<<"1.Load Data Structure\n";

cout<<"2.Print Course List\n";

cout<<"3.Print Course\n";

cout<<"4.Exit\n";

int ch;

// loop will break once user enter 4

do{

// Prompt user to enter choice

cout<<"\nEnter your choice: ";

cin>>ch;

switch(ch)

{

case 1: courses=LoadDataStructure(); break;

case 2: printCourseList(courses); break;

case 3: searchCourse(courses); break;

case 4: cout<<"Exit\n"; break;

default: cout<<"Invalid Choice\n";

}

}while(ch!=4);

return 0;

}