

Course T1Y2: Advanced Algorithms

Lecturer: Bou Channa

Student’s name: Chea Ilong

ID: 100022

Group: 1 SE Gen10

Lab6 assignment-FILE IO

Exercise1:

#include <iostream>

#include <fstream>

#include <sstream>

#include <string>

using namespace std;

struct Book

{

    string ISBN, title, category, authors, price, publishedYear;

    Book \*next;

};

class LinkedList

{

    Book \*head, \*tail;

    int size;

public:

    LinkedList()

    {

        head = NULL;

        size = 0;

    }

    ~LinkedList()

    {

        while (head)

        {

            Book \*temp = head;

            head = head->next;

            delete temp;

        }

    }

    void insertBegin(const string &ISBN, const string &title, const string &category, const string &authors, const string &price, const string &publishedYear)

    {

        Book \*b = new Book{ISBN, title, category, authors, price, publishedYear, head};

        head = b;

        size++;

    }

    void displayList()

    {

        Book \*t;

        t = head;

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

        {

            cout << ". ISBN: " << t->ISBN << ", Title: " << t->title << ", Category: " << t->category << ", Authors: " << t->authors << ", Price: " << t->price << ", Published Year: " << t->publishedYear << endl;

            t = t->next;

        }

    }

    void csvSplit(const string &line, string &ISBN, string &title, string &category, string &authors, string &price, string &publishedYear)

    {

        stringstream ss(line);

        getline(ss, ISBN, ',');

        getline(ss, title, ',');

        getline(ss, category, ',');

        getline(ss, authors, ',');

        getline(ss, price, ',');

        getline(ss, publishedYear, ',');

    }

};

int main()

{

    LinkedList \*mylist = new LinkedList;

    ifstream myfile;

    myfile.open("C:\\Users\\MSI PC\\Desktop\\lab6\\book-list.csv", ios::in);

    if (!myfile.is\_open())

    {

        cout << "Error: Could not open the file!" << endl;

        return 1;

    }

    string line;

    while (getline(myfile, line))

    {

        string ISBN, title, category, authors, price, publishedYear;

        mylist->csvSplit(line, ISBN, title, category, authors, price, publishedYear);

        mylist->insertBegin(ISBN, title, category, authors, price, publishedYear);

    }

    mylist->displayList();

    myfile.close();

    delete mylist;

    return 0;

}

A screenshot of a computer program

Description automatically generated

A screen shot of a computer screen

Description automatically generated

Exercise2:

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct Employee

{

    string name, gender, email, age, salary;

};

int main()

{

    const int n = 10;

    Employee employees[n];

    for (int i = 0; i < n; i++)

    {

        cout << "Enter details for Employee " << (i + 1) << ":" << endl;

        cout << "Name: ";

        cin >> employees[i].name;

        cout << "Age: ";

        cin >> employees[i].age;

        cout << "Gender: ";

        cin >> employees[i].gender;

        cout << "Email: ";

        cin >> employees[i].email;

        cout << "Salary: ";

        cin >> employees[i].salary;

        cout << endl;

    }

    ofstream myfile;

    myfile.open("C:\\Users\\MSI PC\\Desktop\\lab6\\exercise-2.csv", ios::out);

    if (!myfile.is\_open())

    {

        cout << "Error: Could not open the file!" << endl;

        return 1;

    }

    myfile << "Name,Age,Gender,Email,Salary" << endl;

    for (int i = 0; i < n; i++)

    {

        myfile << employees[i].name << ","

               << employees[i].age << ","

               << employees[i].gender << ","

               << employees[i].email << ","

               << employees[i].salary << endl;

    }

    myfile.close();

    return 0;

}

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

Exercise3:

#include <iostream>

#include <fstream>

#include <sstream>

#include <string>

#include <climits>

#include <vector>

using namespace std;

struct Student

{

    string StudentID, Name, Gender;

    int MathScore, EnglishScore, ScienceScore;

};

int main()

{

    vector<Student> studentList;

    ifstream infile("C:\\Users\\MSI PC\\Desktop\\lab6\\data.csv", ios::in);

    ofstream outfile("C:\\Users\\MSI PC\\Desktop\\lab6\\data\_report.csv", ios::out);

    if (!infile.is\_open() || !outfile.is\_open())

    {

        cout << "Error: Could not open the file!" << endl;

        return 1;

    }

    string line;

    getline(infile, line);

    while (getline(infile, line))

    {

        stringstream ss(line);

        Student student;

        string mathScore, englishScore, scienceScore;

        getline(ss, student.StudentID, ',');

        getline(ss, student.Name, ',');

        getline(ss, student.Gender, ',');

        getline(ss, mathScore, ',');

        getline(ss, englishScore, ',');

        getline(ss, scienceScore, ',');

        student.MathScore = stoi(mathScore);

        student.EnglishScore = stoi(englishScore);

        student.ScienceScore = stoi(scienceScore);

        studentList.push\_back(student);

    }

    int totalStudents = studentList.size();

    int maleCount = 0, femaleCount = 0;

    int maxScore = 0;

    int minScore = INT\_MAX;

    int totalscore;

    for (const auto &student : studentList)

    {

        if (student.Gender == "M")

        {

            maleCount++;

        }

        else if (student.Gender == "F")

        {

            femaleCount++;

        }

        totalscore = student.MathScore + student.EnglishScore + student.ScienceScore;

        if (totalscore > maxScore)

        {

            maxScore = totalscore;

        }

        if (totalscore < minScore)

        {

            minScore = totalscore;

        }

    }

    outfile << "Total Students:" << totalStudents << endl;

    outfile << "Males:" << maleCount << endl;

    outfile << "Females:" << femaleCount << endl;

    outfile << "Max Score:" << maxScore << endl;

    outfile << "Min Score:" << minScore << endl;

    infile.close();

    outfile.close();

    cout << "Data report has been written to data\_report.csv" << endl;

    return 0;

}

Read the file from

A computer screen shot of a computer program

Description automatically generated

Then write in

A screenshot of a computer

Description automatically generated