```
/* ----------------------------------------------------------------
// student.h
// ---------------------------------------------------------------- */
#ifndef student_HHH
#define student_HHH
#include <iostream>
#include <string>
class Student{
    friend std::ostream& operator << (std::ostream& os, const Student &stu);
    public:
        bool operator < (const Student &rhs) const;
        void Input(std::istream& = std::cin);
        const char* GetName() const;
    private:
        static constexpr int score_num_ = 3;
        static constexpr int size_name_ = 31;
        char name_[size_name_] = {'\0'};
        char gender_ = 'M';
        int age_ = 0;
        int scores_[score_num_] = {0};
};
std::ostream& operator << (std::ostream& os, const Student &stu);
#endif
/* ----------------------------------------------------------------
// student.cpp
// ---------------------------------------------------------------- */
#include "student.h"
using namespace std;
ostream& operator << (std::ostream& os, const Student &stu){
    os << stu.name_ << " " << stu.gender_ << " ";
    if(os == cout){
        os << "(" << stu.age_ << ")";
    }else{
        os << stu.age_;
    }
    for(int i=0;i<stu.score_num_;i++)
        os << " " << stu.scores_[i];
    os << endl;
```

```
    return os;
}

void Student::Input(istream& is){
    if(is == cin){
        cout << "Name: ";
        is >> name_;
        cout << "Gender(M/F): ";
        is >> gender_;
        cout << "Age: ";
        is >> age_;
        cout << score_num_ << " Scores:";
        for(int i=0;i<score_num_;i++)
            is >> scores_[i];
    }else{
        is >> name_ >> gender_ >> age_ ;
        for(int i=0;i<score_num_;i++)
            is >> scores_[i];
    }
}

const char* Student::GetName() const{
    return name_;
}

bool Student::operator < (const Student &rhs) const {
    string s1 = name_;
    string s2 = rhs.name_;
    return s1 < s2;
}
/* ----------------------------------------------------------------
// database.h
// ---------------------------------------------------------------- */
#ifndef database_HHH
#define database_HHH
#include "student.h"
#include <iostream>
#include <fstream>
```

```
#include <algorithm>
#include <string>
class Database{
    friend std::ostream& operator << (std::ostream& os, const Database &db);
    public:
        void Add(const Student&);
        void Search(std::string)const;
        bool isFull()const{ return (size_ >=DB_MAX_SIZE); }
        bool ReadFile(std::string);
        bool OutputFile(std::string)const;
        int GetSize()const{return size_;};
        void sort();
    private:
        static constexpr int DB_MAX_SIZE = 100;
        Student students_[DB_MAX_SIZE];
        int size_ = 0;
};
std::ostream& operator << (std::ostream& os, const Database &db);
#endif


/* ----------------------------------------------------------------
// database.cpp
// ---------------------------------------------------------------- */

#include "database.h"
using namespace std;

ostream& operator << (ostream& os, const Database &db){
    for(int i=0;i<db.size_;i++)
        os << db.students_[i];
    return os;
}

void Database::Add(const Student& stu){
    students_[size_++] = stu;
}
```

```
bool Database::ReadFile(string filename){
    Student temp;
    ifstream ifile(filename);
    if(ifile){
        size_ = 0;//Reset the size of the Database
        while(ifile){
            if(isFull()){
                cout << "The database is full. ";
                return false;
            }
            temp.Input(ifile);
            if(ifile)
                Add(temp);
        }
        ifile.close();
        return true;
    }else{
        return false;
    }
}

void Database::sort(){
    stable_sort(this->students_,this->students_+size_);
}

void Database::Search(string name)const{
    int result_ind[DB_MAX_SIZE] = {0};
    int count = 0;
    for(int i=0;i<size_;++i){
        if( name == this->students_[i].GetName() )
            result_ind[count++] = i;
    }
    cout << endl << "There is/are " << count << " \"" << name << "\"" << endl
<< endl;
    for(int i=0;i<count;++i)
        cout << this->students_[result_ind[i]];
}
```

```
bool Database::OutputFile(string filename)const{
    ofstream ofile(filename);
    if(ofile){
        ofile << (*this);
        ofile.close();
        return true;
    }else{
        return false;
    }
}


/* ----------------------------------------------------------------
// main.cpp
// ---------------------------------------------------------------- */


#include "database.h"
using namespace std;
enum Option{Add = 1, Show, Search, OutputFile, InputFile, Exit};
int main(){
    int opt = 0;
    char select = '\0';
    Database db;
    Student stu;
    string name;
    do{
        cout << "A simple database program" << endl << endl
            << "(1)Add a record" << endl
            << "(2)Show all records" << endl
            << "(3)Search student(s) by name" << endl
            << "(4)Output records to a text file" << endl
            << "(5)Input records from a text file" << endl
            << "(6)Exit the program" << endl << endl;
        cout << "Please select a function...>";
        cin >> opt;
        system("pause");
        system("cls");
```

```
        switch(opt){
            case Add:
                cout << "Please input the required values:" << endl << endl;
                if(db.isFull()){
                    cout << "The database is full, so you can't add more records.";
                }else{
                    stu.Input();
                    db.Add(stu);
                }
                break;
            case Show:
                if(db.GetSize() > 1)
                    cout << "There are " << db.GetSize() << " records:" ;
                else
                    cout << "There is " << db.GetSize() << " record:" ;
                cout << endl << endl << db;
                break;
            case Search:
                cout << "Please input the name of the student...>";
                cin >> name;
                db.Search(name);
                break;
            case OutputFile:
                cout << "Out all records to a text file." << endl << endl
                    << "Note that the original data in the file will be lost after
writing to the file. "
                    << "Are you sure that you want to continue? (Y/N)...>";
                cin >> select;
                cout << endl;
                if(toupper(select) == 'Y'){
                    cout << "Please input the name of the file...>";
                    cin >> name;
                    if(!db.OutputFile(name))
                        cout << "failure to output file." << endl;
                }
                break;
            case InputFile:
                cout << "Read in records from a text file." << endl << endl
```

```
                << "Note that current data in the program will be lost after
reading. "
                << "Are you sure that you want to continue? (Y/N)...>";
            cin >> select;
            cout << endl;
            if(toupper(select) == 'Y'){
                cout << "Please input the name of the file...>";
                cin >> name;
                if(!db.ReadFile(name))
                    cout << "failure to input file." << endl;
            }
            break;
        case Exit:
            return 0;
        default:
            cout << "Invalid Command";
    }
    db.sort();
    cout << endl << endl << "Press ENTER to continue.";
    system("pause");
    system("cls");
}while(1);
return 0;
}
```