Data Structures and Algorithms Lab

Lab 02 **Marks 05**

Instructions

Work in this lab individually. You can use your books, notes, handouts etc. but you are not allowed to borrow anything from your peer student.

Marking Criteria

Show your work to the instructor before leaving the lab to get some or full credit.

Task 01

Write a function named **arrayToFile**. The function should accept three arguments: the **name** of a file, a **pointer** to an **int array**, and the **size** of the array. The function should **open** the specified file in **binary mode**, **write** the contents of the array to the file using a **single statement** without any looping, and then **close** the file.

Write another function named **fileToArray**. This function should accept three arguments: the **name** of a file, a **pointer** to an **int array**, and the **size** of the array. The function should **open** the specified file in **binary mode**, **read** the required number of elements based on the **size** provided as an argument into the array, and then **close** the file. *The function should not display anything.*

Demonstrates the working by using the **arrayToFile** function to **write** an array to a file, and then using the **fileToArray** function to **read** the data from the same file. After the data is read from the file into the array, **display** the array contents on the screen in **main** function.

Task 02

```
class Student
{
        int id;                         //store the id of a student
        float marks;                    //store the marks of a student

        //overload stream extraction operator to get data of a student from user
        friend istream & operator >> (istream &, Student &);

        //overload stream insertion operator to display the data of a student
        friend ostream & operator << (ostream &, const Student &);

public:
        //constructor
        Student(int id = 0, float marks = 0.0f);
};
```

Provide the implementation the following member functions (instance or static) for the **Student** class;

addStudent – An **instance member function** which should accept the reference of **ofstream** as its argument, write the data of left-hand side object into the file (**student.dat)** pointed by the **ofstream's** object and give confirmation message.

displayAllStudents – A **static member function** which should accept the reference of **ifstream** as its argument, **display all the records** exist in the file **(student.dat)** pointed by the **ifstream's** object in the proper format otherwise display a **proper message** if the **file is empty or does not exist**.

findStudent – An **instance member function** which should accept the reference of **ifstream** as its argument, return true if the file **(student.dat)** pointed by the **ifstream's** object contains data equals to the left-hand side object or display a **proper message** if the **student does not exist**.

deleteStudent – An **instance member function**, which should accept the reference of **ifstream** as its argument, **delete** the **student** exist in the file **(student.dat)** pointed by **ifstream's** object having same information as of the left-hand side object or display a **proper message** if the **student does not exist**.

Implement **main** function and create/open **student.dat** file in it. Test the functionality of **Student** class by creating some of its objects. Write these objects in **student.dat** file by making calls to appropriate member functions. Provide a menu through which the user can add, find, delete, or display the records to/from the file.

☺ ☺ ☺ **BEST OF LUCK** ☺ ☺ ☺