



Fayoum University  
Faculty of Computers & Information  
Second Term 2019/2020

Course Name: File Organization

Course Code: INF271

Course Level: 2<sup>nd</sup>

Course Dr. Name : Howida Youssry

Status: Pass  Not Pass:

**Choose one of the following topics for your research:**

**Topic 1:**

Implement an **Employees Management System**, using C++, which stores and retrieves employee data. Your system MUST implements the following functionalities:

Employee Management System

1. Add New Employee
2. Update an Employee
3. Delete an Employee
- 4) Display all Employees
- 5) Display all Employees in specific Department
- 6) Search Employee by Serial
- 7) Compress File
- 8) Exit

Consider we have data of a set of employees; in which every employee has the following attributes:

- Serial : char [5] // Unique Serial number contains characters and digits:  
ex: 'A247'
- First Name : char [10] //max len = 10
- Last Name : char [10] //max len = 10
- Age: short
- Department: char [10] //max len=10
- Salary: float

## Topic 2:

Implement a **Students Management System**, using C++, which stores and retrieves student data. Your system MUST implements the following functionalities:

### Students Management System

1. Add New Student
2. Update Student
3. Delete Student
4. Display All Students
5. Search by ID
6. Search by Name
7. Clean File
8. Exit

Consider we have data of a set of students; in which every student has the following attributes:

- Name: char [20]      //max len = 20
- ID : int              // Unique ID
- GPA: float
- Address: char [50]    //max len=50

### Topic 3:

Implement **Product Management System**, using C++, which stores and retrieves product data. Your system MUST implements the following functionalities:

#### Product Management System

1. Add New Product
2. Update Product
3. Delete Product
4. Display All Product
5. Search by ID
6. Search by Name
7. Clean File
8. Exit

- Assume you have records of Products having the following 5 attributes:
  - int id; // Unique ID
  - char name[20]; //max len = 20
  - char category[20];
  - float price;
  - char inStock; // Y or N

#### Topic 4:

Implement **Hospital Management System**, using C++, which stores and retrieves product data. Your system MUST implements the following functionalities:

Product Management System

9. Add New patient
10. Update patient
11. Delete patient
12. Display All patient
13. Search by ID
14. Search by Name
15. Clean File
16. Exit

- Assume you have records of Products having the following 5 attributes:
  - int id; // Unique ID
  - char patient\_name [20]; //max len = 20
  - char disease [40];
  - char patient\_type[20];
  - float age;

- يختار الطالب احد الموضوعات عاليه لاجراء مقالة بحثية
- يكتب البحث الكترونياً ويرسل كملف واحد بصيغة PDF على البريد الالكتروني لاستاذ المقرر عاليه على ان يسمى الملف وكذلك موضوع البريد الالكتروني باسم الطالب ثلاثى والمقرر.
- سيتم رفض البحوث المتشابهة جزئيا او كليا ويعتبر الطلاب المتقدمون بها راسبين فى هذه المادة
- الموضوعات المطروحه هى لما تم تدريسه سواء عن بعد او قبل تعليق الدراسة.

## Employee Management System (Case Study)

Consider we have data of a set of employees; in which every employee has the following attributes:

- Serial : char [5] // Unique Serial number contains characters and digits: ex.'A247'
- First Name : char [10] //max len = 10
- Last Name : char [10] //max len = 10
- Age: short
- Department: char [10] //max len=10
- Salary: float

Implement an **Employees Management System**, using C++, which stores and retrieves employee data. Your system MUST implements the following functionalities:

Employee Management System

4. Add New Employee
5. Update an Employee
6. Delete an Employee
- 9) Display all Employees
- 10) Display all Employees in specific Department
- 11) Search Employee by Serial
- 12) Compress File
- 13) Exit

Please Enter Your Choice:

### 1. Add New Employee:

- Add new employee data taken from the user.
- Checks first the Avail-List using the **FIRST FIT** strategy, if you can use a deleted slot then add the new record in it, if no deleted slot is sufficient (i.e. size of all deleted slots are smaller than the size of the new record) then add the new record to the end of the file.
- Add new record at both indices : primary and secondary
- Keep primary index sorted after the addition of the new record

### 2. Update Employee:

- Update an employee record given its unique serial.

- If the new record size is smaller than or equal to the old record, then the updated record should be placed in the same place.
- If the new record size is larger, then perform delete operation then add operation.
- Update the department in the secondary index.

**3. Delete Employee:**

- Delete an employee record given its unique serial.
- Mark the employee record in the data file ONLY as deleted '\*' and refer to the next deleted record.
- Keep the secondary department index as it is (No change).
- Remove the record from the primary index.
- Update the Avail-List after each deletion by adding the newly deleted record to it on the data file.

**4. Display All Employees:**

- Displaying all employee records (Non-deleted ONLY)
- Use the primary index to display all records, and since your primary index is sorted ascendingly, then the records displayed should be sorted by Serial as well.

**5. Display all Employees in a Specific Department**

- Search for all employees that are working in a given department.
- If department found, then display the employee(s) data, otherwise printing that this department is not found.
- If more than an employee matched with the given department, they should be displayed ALL.

**6. Search Employee by Serial:**

- Search for an employee given its unique serial.
- If serial found, then display the employee data, otherwise print a message that the given serial is not found.

**7. Compress File:**

- Deletes physically all the deleted records from: data and the secondary index.
- The size of the files should become smaller after compressing.
- The primary index should be updated by the new offsets.

**8. Exit:** Should exit your system.

➤ Some requirements and notes that you MUST follow in your system:

- Have three physical files :
  - Data file : contains all Employee record fields [ Serial, First Name, Last Name, Age, Department, and Salary]

- Primary Index file: contains the employee Serial, and record offset (starting byte at the data file).
- Secondary Index file: contains the employee Department, and Serial.
- The Serial of ANY employee consists of a character followed by three numeric digits; For example: A123, D456, A009 ... etc.
- The Avail-List must be implemented in the Data file, so a **2-byte** header field should be added at the beginning of your **data** file.
- When updating an employee, any field of employee data can be updated except the Serial it should remain the same.
- Bind Secondary Index with primary key (Serial), not with record offset.
- The primary index **MUST** be sorted ascendingly all the time.
- Use the primary Index to search for an employee by its unique Serial, and use the secondary Index to search for all employees in a department.
- When searching by Department, search first the secondary index and get the Serial then search the primary index with the matched Serial. If the Serial is not found in the primary index, then its record should be removed from the secondary index at the moment.
- The two indices should be loaded into memory (in arrays) at system start and written back to files at system exit.
- **The structure of the three files are:**
  - Data file: **variable length-based record** with **variable length-based field**, and starting with 2-bytes for avail-list header followed by records,
    - **record\_size(2-bytes) field\_size(2-bytes) Serial (5-bytes) field\_size(2-bytes) First Name field\_size(2-bytes) Last Name field\_size(2-bytes) Age (2-bytes) field\_size(2-bytes) Department field\_size(2-bytes) Salary (4-bytes)**
  - Primary Index file: fixed-length fields (7 bytes per record)
    - records (5 bytes for Serial + 2 bytes for offset)
  - Secondary Department Index file : fixed-length fields (15 bytes per record)
    - records (10 bytes for Department + 5 bytes for Serial)

➤ **PAY ATTENTION:**

You must follow the exact file structure mentioned in this document for the three files because at discussion time your system will be tested on external data and indices files prepared by the TAs. So if you implemented your system on a different file structure, it will not run correctly on the test cases files.



➤ **Test cases files provided:**

- There are 3 files provided with this document to test your work on them:
  - *Employees.txt* → The data file contains data of 6 records added, a record with Serial =A201 and a record with Serial =A214 were deleted, so the avail-list on this file contains 2 records.
  - *prim-index.txt* → The primary index file, contains the records of the 4 non-deleted records ONLY
  - *scnd-index.txt* → The secondary name index, contains the records of the 6 records. If you performed search on Department “Marketing”, the corresponding deleted records with serials “A214” & “A201” should be removed from the secondary index.
- Try hard to make your code work fine with those files.
- The following is a simple run on the files provided:

```
Employee Management System
1) Add New Employee
2) Update an Employee
3) Delete an Employee
4) Display All Employees
5) Display all Employees in specific Department
6) Search Employee by Serial
7) Compress File
8) Exit
Please Enter Your Choice: 4

Serial   First Name   Last Name   Age   Department   Salary
A001     Ahmed       Ali         50    Testing      3000
A201     Maha        Mokhtar    35    HR            2500
A210     Amgad       Mohamed     48    IT            4000
A214     Amal        Sabry       36    Testing      3500
B105     Kareem      Maged       30    HR            2500
S301     Omar        Mohamed     46    Account      4500
```

Employee Management System

- 1) Add New Employee
- 2) Update an Employee
- 3) Delete an Employee
- 4) Display All Employees
- 5) Display all Employees in specific Department
- 6) Search Employee by Serial
- 7) Compress File
- 8) Exit

Please Enter Your Choice: 3

Enter the Employee Serial to delete: A201

Employee Management System

- 1) Add New Employee
- 2) Update an Employee
- 3) Delete an Employee
- 4) Display All Employees
- 5) Display all Employees in specific Department
- 6) Search Employee by Serial
- 7) Compress File
- 8) Exit

Please Enter Your Choice: 3

Enter the Employee Serial to delete: A214

Employee Management System

- 1) Add New Employee
- 2) Update an Employee
- 3) Delete an Employee
- 4) Display All Employees
- 5) Display all Employees in specific Department
- 6) Search Employee by Serial
- 7) Compress File
- 8) Exit

Please Enter Your Choice: 4

Serial	First Name	Last Name	Age	Department	Salary
A001	Ahmed	Ali	50	Testing	3000
A210	Amgad	Mohamed	48	IT	4000
B105	Kareem	Maged	30	HR	2500
S301	Omar	Mohamed	46	Account	4500