

## A Sample for a Report in subjects: OOP using Java, Desktop Java Application, Web-based Java Application, C# and Dot Net

**Reports are written in your file.**

### 1- Problem Description

- Each employee details include: code, name, salary
- The text file, named employees.txt contains some initial employee details in the following line-by-line format: code, name, salary
- Write a Java program having a simple menu that allows users managing a list of employees. Functions are supported:
  - Adding new employee
  - Removing employee.
  - Promoting the salary of an employee.
  - Listing employee details.
  - Save the list to file
  - Quit

### More requirements

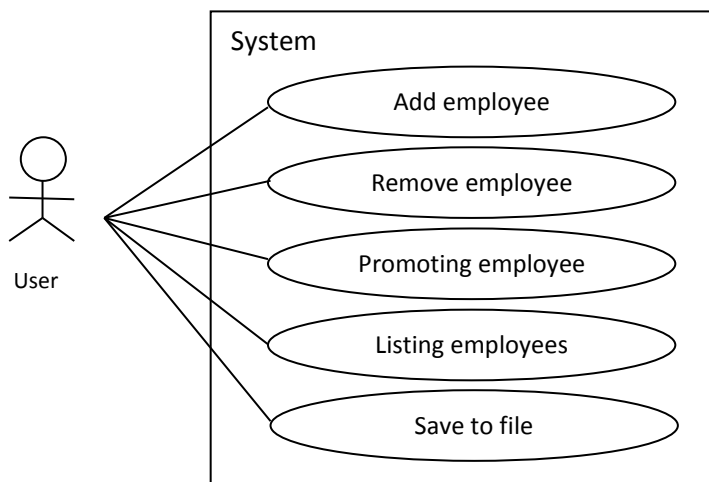
Format of the code of employee: E000, in which 0s are digits

Name of employee is uppercase.

The printed list have order of ascending codes.

### 2- Analysis

From the problem description, the following use-cases are identified:



### 3- Design

## Class Design

From the main concepts in the problem, following classes are identified:

Concept	Class
Employee (code, name, salary)	Employee (code, name, salary)
List of employees	EmpList, a collection of employees
Menu	Menu, a collection of strings
Program	ManageProgram

### Class details

Menu: Vector<String>
+ addMenuitem(String s): void + getUserChoice(): int

Employee
- code: String - name: String - salary: int + Employee(String, String, int) + getters/ setters + compareTo(Object emp): int + print()

EmpList: Vector<Employee>
+ EmpList() + addFromFile(String fname): void + addNewEmp(): void - find (String aCode): int + print(): void + promote(): void + removeEmp(): void + saveToFile(String fname): void

ManageProgram
+ main(String[]): void

( +: public member, -: private member)

### User Interface:

Menu of the program will be seen as:

- 1- Add new employee
- 2- Remove an employee
- 3- Promoting the employee's salary
- 4- Print the list
- 5- Save to file
- 6- Quit

## 4- Implementation

### 1- Initial data of the program ( file employees.txt)

Please explore the software structure

### 2- Software

Please run the program

## 5- Testing

Test	Result (passed/ not passed)
Format of the code of an employee: E000	Passed
Name of employees are uppercase	Passed
List is printed as an ascending order of codes	Passed

**6-task list and evaluation:** Details and evaluation task list and contribution for each member

## **Requirements:**

### Code Conventions:

- Indentation: 4 blanks at the beginning of each code line
- Comments in the code must be carried out.
- Names:
  - One-word name: lowercase
  - Multi-word name: The first word: lowercase, remaining words: The first character is uppercase, others are lowercase.

A sample :

```
/*
  Author: .....
  Date: .....
  This class represents .....

*/
class ClassName ..... {
    int data; // Which does data represent?
    ...
    /* What is the goal of the method
       Which does the return data represent?

    */
    Method implementation .... {
    ...
    }
}
```