

## Objective

The purpose of this assignment is to write a program that uses loops, file I/O, and methods to process employee salary data and generate a small report. Additionally, you will gain further practice with several concepts that you have already worked with including: strings and string-handling functions; console I/O; and branching. Use of arrays and/or ArrayLists are not allowed.

## Instruction

Your team has been asked to develop a program that accepts two data filenames and an employee name (first & last name) as user input, searches both files for the specified employee's information, and optionally prints a formatted salary schedule for that employee. Separate static methods should be used in your program, as described further in the Solution Design section.

## Input Files

Both input files are plain text files in tab-delimited format. The first line of each file is a header that can be skipped over by your program.

Here are a few lines extracted from each input file to show the layout.

### Employee.txt:

EmpLastName	EmpFirstName	BirthDate	ServiceStartDate	EmpID
Abbott	Zoe	4/10/1969	6/25/2020	E55049
Adams	Amaya	12/30/1968	5/29/2020	E55042
...				
Bauer	Dolan	6/26/1960	1/5/2018	E55001
...				

### EmpSal2022.txt:

EmpID	MonthlySalary	FedTax	ABTax	CPP	EI	OptRSP
E55001	5600	681.85	365.8	289.3	88.48	280
E55002	10000	1679.45	812.5	529.07	158	200
...						

Information in the two files is related and the common element, Employee ID or EmpID, is what ties the data together. For example, looking first at the Employee.txt sample above, we can find out that the EmpID for Dolan Bauer is E55001. We can then locate Dolan's salary information in the EmpSal2022.txt file from the line with the matching EmpID E55001, which shows:

- 5600 – gross monthly salary
- 681.85 – federal monthly income tax
- 365.8 – Alberta monthly income tax
- 289.3 – monthly CPP payment, up to the maximum of 3,499.80 per year in 2022
- 88.48 – monthly EI payment, up to the maximum of 952.74 per year in 2022
- 280 – monthly optional retirement savings plan (RSP) contribution

## CMPP269 Assignment #2 Winter 2022 – Employee Salary Program

### Test Plan

Below are 4 separate sample runs which make up the test plan and demonstrate how your program should obtain user input and present results to the user. User input appears as **red bold underline**:

#### Sample run #1:

```
Enter employee filename (full path): c:/temp/employee.txt
Enter employee first name: Dolan
Enter employee last name: Bauer

Employee information is found for: Dolan Bauer
And employee ID is: E55001
*****

Would you want to retrieve the employee salary information (Y/N): Y
Enter employee salary filename (full path): c:/temp/empsal2022.txt

-----
Salary information for DOLAN BAUER with E55001 is:
-----
      Monthly Salary:  $5600.00
Monthly Federal Tax:   $681.85
Monthly Alberta Tax:  $365.80
      Monthly CPP:    $289.30
      Monthly EI:     $88.48
    Monthly Opt. RSP:  $280.00
*****

Would you want to calculate deductions and print salary schedule/year (Y/N): Y
```

Salary Schedule per year:

Month	Gross Salary	Tax	CPP	EI	Optional RSP	Net Salary
1	5,600.00	1,047.65	289.30	88.48	280.00	3,894.57
2	5,600.00	1,047.65	289.30	88.48	280.00	3,894.57
3	5,600.00	1,047.65	289.30	88.48	280.00	3,894.57
4	5,600.00	1,047.65	289.30	88.48	280.00	3,894.57
5	5,600.00	1,047.65	289.30	88.48	280.00	3,894.57
6	5,600.00	1,047.65	289.30	88.48	280.00	3,894.57
7	5,600.00	1,047.65	289.30	88.48	280.00	3,894.57
8	5,600.00	1,047.65	289.30	88.48	280.00	3,894.57
9	5,600.00	1,047.65	289.30	88.48	280.00	3,894.57
10	5,600.00	1,047.65	289.30	88.48	280.00	3,894.57
11	5,600.00	1,047.65	289.30	67.94	280.00	3,915.11
12	5,600.00	1,047.65	289.30	0.00	280.00	3,983.05
Total	\$67,200.00	\$12,571.80	\$3,471.60	\$952.74	\$3,360.00	\$46,843.86

#### Sample run #2:

```
Enter employee filename (full path): c:/temp/employee.txt
Enter employee first name: orson
Enter employee last name: avery

Employee information is found for: orson avery
```

## CMPP269 Assignment #2 Winter 2022 – Employee Salary Program

And employee ID is: E55059

\*\*\*\*\*

Would you want to retrieve the employee salary information (Y/N): n  
Bye...

### Sample run #3:

Enter employee filename (full path): c:/temp/employee.txt

Enter employee first name: Kay

Enter employee last name: Woodward

Employee information is found for: Kay Woodward

And employee ID is: E55035

\*\*\*\*\*

Would you want to retrieve the employee salary information (Y/N): y

Enter employee salary filename (full path): c:/temp/empsal2022.txt

-----  
Salary information for KAY WOODWARD with E55035 is:  
-----

Monthly Salary:	\$4700.00
Monthly Federal Tax:	\$504.15
Monthly Alberta Tax:	\$276.10
Monthly CPP:	\$240.25
Monthly EI:	\$74.26
Monthly Opt. RSP:	\$0.00

\*\*\*\*\*

Would you want to calculate deductions and print salary schedule/year (Y/N): n  
Bye...

### Sample run #4:

Enter employee filename (full path): c:/temp/employee.txt

Enter employee first name: Musk

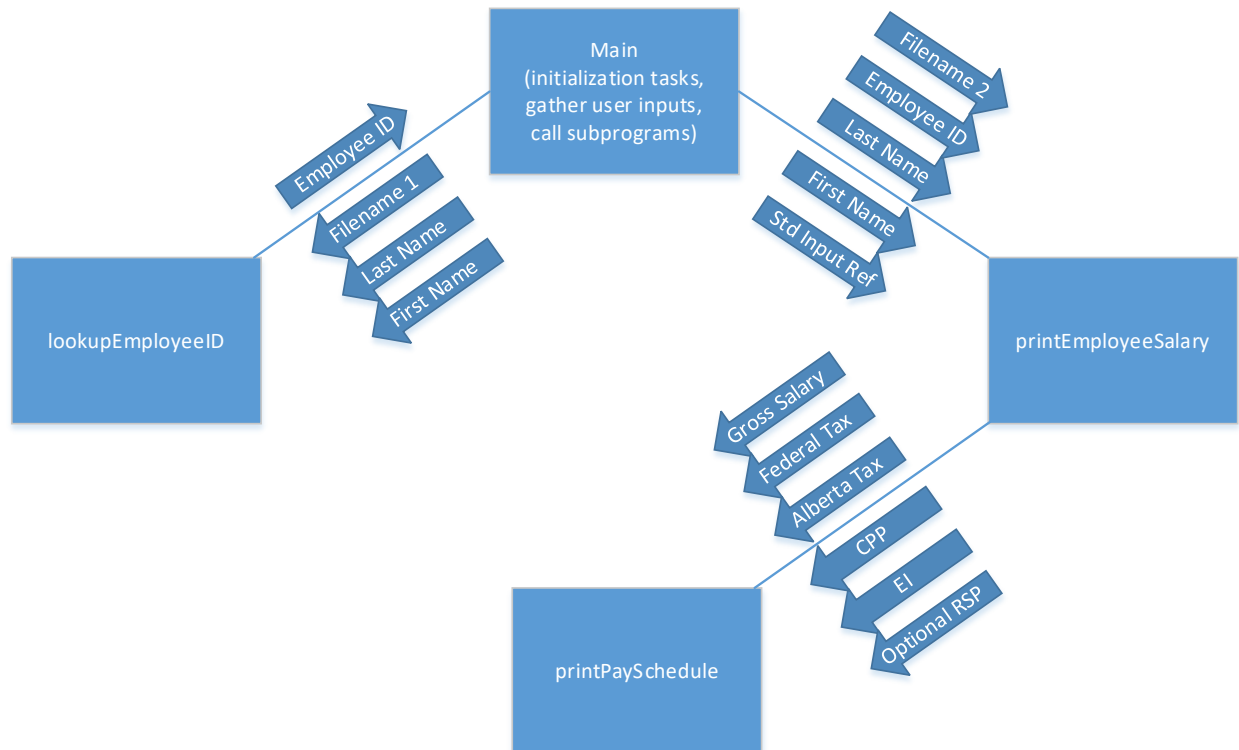
Enter employee last name: Kay

No employee information is found for: Musk Kay

## Solution Design

A high level design for the program has been done and your team is expected to adhere to this design, details of which are given below.

Structure chart:



Additional design considerations:

- Use the procedural programming approach (i.e. what has been used in this course so far) rather than an object-oriented approach.
- This program will find and print salary information for a single employee only. In other words the input files will only be read once per execution of the program and you will NOT need to use arrays or other more sophisticated Java data types.
- Keep file I/O code within the appropriate method (i.e. code to process Employee.txt in the lookupEmployeeID method and code to process EmpSal2022.txt in the printEmployeeSalary method).
- Use the Scanner class to read the files as shown in class and in the course material.

## Group Development

Since this assignment will involve writing multiple methods, it is recommended that your group initially work out what the parameters and return values need to be, using the information provided in the solution design section. A great start would be to define the headers for all of the methods. Then you could divide the programming tasks among your team members and even make use of the "stub" development technique. Once the methods have been written the modules can be put together and then the full program tested and refined.

## Submission and Grading

Please follow the specific submission requirements from your instructor. For a group assignment this would generally mean that one solution, consisting of your group's final Java source code and accompanying text file with output from the test runs, be submitted to Brightspace.

### **Submissions will be evaluated according to the following standard:**

1. Working Java code (60%)
  - Follows the high level design provided
2. Style (20%)
  - Indentation – consistent
  - Readability – good variable names
  - Documentation
    - Comments at the top which include: Name, date, and program description including details on inputs, processing and outputs (4-5 sentences minimum).
    - Block comments indicating major code sections and what they do. Including a comment block just before each method is good practice!
3. A sample run that matches the provided test plan (20%)
  - Output formatted according to specification (test plan)