**Practical**

**(Junior/Intermediate)**

**Basic CSV File operations**

**Table of Contents**

[**1**](#_30j0zll) **Document Information 3**

[**1.1**](#_1fob9te) **Document History 3**

[**1.2**](#_3znysh7) **Introduction 4**

[1.2.1](#_2et92p0) Junior 4

[**1.3**](#_tyjcwt) **Intermediate 6**

[1.3.1](#_3dy6vkm) Requirements: 6

# Document Information

## Document History

| Version | Date | Editor | Changes Made |
| --- | --- | --- | --- |
| 0.1 | 2022-01-13 | T Claassens | Initial |

## Introduction

Please see 2 sections below. Junior and Intermediate both with a bonus question. Please CHOOSE a section you are most comfortable with and Complete. If you complete the Junior Section please see the Intermediate Section as additional bonus questions. Please Note, the Bonus questions are not mandatory.

### Junior

#### Module 1.1 - Basic CSV Parse and Summary Generation

**Overview:**

This practical will be covering basic file operations on a CSV file, including writing a summary file to a specific location.

#### Requirements:

#### Description:

Consume a CSV file that contains Salary Information and Generate a Summary File in CSV format.

#### Estimated Completion time:

12 Hours

#### Requirements:

* Application must be a spring boot app.
* Application must be able to run from a command line argument.
* Offer ability to input a file path as a command line argument
* Parsing of Data to Java Objects
* Offer ability to input the result file path and name as a command line argument
* Offer ability to choose separator used for the input file on the command line
* Application must be Java Maven project
* For Parsing of the CSV file, No 3rd party APIs. Pure Java Only

**Input File** : The input file is called *Employee.csv*

**Example** :



Output file Format :

Requirements:

* CSV File (Comma Delimited)
* Summary per Department detailing the following :

1. Amount of employees
2. Minimum Salary paid in the department
3. Average of the salaries for the department
4. Maximum Salary paid in the department
5. Total of all the salaries in the department
6. Last line of the file needs to list a “Department” called Totals where the totals of all the values are present

* The Headers for the file needs to be as follows:

1. Department
2. Employees
3. MinSalary
4. AvgSalary
5. MaxSalary
6. TotalSalary

**(NB: The Departments must be in Alphabetical order)**

Output of the file needs to look similar to this(Note this example is incomplete. Not all departments are listed and values for the totals are dummy values :



#### BONUS Question (Junior)

1. Please use docker to spin up a db of your choice and Persist Above Summary to the DB Using JDBC

## Intermediate

**Overview:**

Create a spring boot web based application that can return a List of Country names that use a certain Language. This application should also have an added web service, that should be able to calculate calculate a Pythagorean Triplet for a specified sum.

### Requirements:

#### Description:

Create a Spring Boot app that will expose 2 Web Service Methods.

#### Requirements:

1. Create a Java + Spring Boot application.
2. Expose a web service that will take a specific Language name, and return the List of Countries that use it. The Web service needs to offer this functionality via SOAP AND REST (So an endpoint for each must be available). As an example , when passing in the Language name “English” , I want to be able to see a list of the Countries that use it.

To achieve this, use the following public SOAP service :

<http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso?wsdl>

1. A Pythagorean triplet is a set of three natural numbers, *a* < *b* < *c*, for which,

*a*2 + *b*2 = *c*2

For example, 32 + 42 = 9 + 16 = 25 = 52

There exists exactly one Pythagorean triplet for which *a* + *b* + *c* = 1000

Expose a web service (REST) to calculate the values of ‘a’,’b’, ‘c’ by passing in the SUM of ‘a’.’b’,’c’ in the Request. Please add comments so we can understand your Thinking Process.

1. Create a DB Instance through Docker and log all incoming requests and responses. Please note the actual Requests need to be logged. In the case of the SOAP requests, that means the entire Soap Envelope.
2. Db Interactions must be through JDBC.
3. Logging added with Log4j2
4. External properties file to contain external call details. .

#### Bonus Question (Intermediate)

1. Provide an Interface Spec for above Web Service.