**Swiss Re Assignment Design Notes**

# **Assignment:**

* Assignment is about loading **BIG COMPANY** organizational reporting structure starting with CEO till the last employee from a CSV file
* Each employee record has following structure {employeeId, lastName, firstName, salary, managerId}, Where manager is also an employee whose data record should also be in the CSV file

# **Rules:**

* **Manager’s Earning Rule**: A manager’s ideal salary earning should be >= 20% and should be <=50% of average of direct subordinates
* **Hierarchy Depth Rule**: Hierarchy depth between the CEO and any employee should be <= 4

# **Queries:**

* List all earning violations for Manager’s Earning Rule
* List all hierarchy depth violations for Hierarchy Depth Rule

# **Key Points:**

* Use Java SE (any version), and JUnit (any version) for tests.
* Use maven for project structure and build
* Application should read data from a file and print out output to console. No GUIs needed.
* Code will be assessed on correctness, simplicity, readability and cleanliness

# **Assumptions:**

* Data file should present in the root directory with name "data.csv"
* Data can present in any order
* There is only one CEO with NO manager ID
* Each line should have all the fields separated by commas in the type expected for that field
* There may be a possibility of using a real database for data storage

# **Architecture:**

* **BigCompany (Main Class)**: This is the class which has main method and start the application
* **BigCompanyController (Controller)**: This is the class that acts as a singleton facade for data read and other business methods mentioned in Queries section. Once the data is available then multiple clients can connect to this singleton facade object and make calls. Since all the methods are “read only” there would be no synchronization issues.
* **BigCompanyEmployeeProvider (Data Provider)**: This is the interface that abstracts data provider source details. An object of this interface would be passed to the Controller object and the Controller uses that object to load data and apply business filters
* **BigCompanyEmployeeFromCsvFile (CSV File Data Provider):** Implements Data Provider interface and loads data exposes data from CSV file
* **BigCompanyEmployeeFromMockObject**: Implements Data Provider interface and mocks employee data
* **BigCompanyException & BigCompanyLogger** : Exception and Logger wrappers