Coding Challenge - Parallel Streams

Problem Statement 1

Concepts

- ➔ Streams
- ➔ Parallel Streams

Goal

To understand the difference between ordinary streams and Parallel streams.

Problem Statement 1

- ❖ A Gaming Application needs to generate random numbers. The random numbers need to be generated continuously based on the options start and stop
- ❖ Use the Stream features to generate the random numbers.

Problem Statement 2

Given

```
public class Employee {

   private String name;
   private int age;
   private BigDecimal salary;

}
```

Using the above employee class do the following operations

- Create a list of employees.
- From the list of employees, you need to filter all the employees whose age is greater then 20 and print the employee names.
- From the list of employees, count the number of employees with an age 25?
- From the list of employees, find the employee with the name “Mary”.
- From the list of employees, find the maximum age of the employee?
- From the list of employees, sort all the employees on the basis of age?

Problem Statement 3

We need to generate the Employee object instance (use the Same Employee class as in Problem statement 2).
(Create this as a separate project)

Write a Java code To generate 10,000 random employee data and save the same using

1. Normal Stream
2. Parallel stream

Save the Employee objects in a separate file. (Each Employee will be saved on a separate file.)
Observe the time taken for 1,2

Sample data

Example

Employee Sam will be save in a file known as **c:\employee**\sam.txt
Employee John will be saved in a file known as **c:\employee**\john.txt

Note: use System.currentTimeMillis() for finding the system time