Context: You are working on a project that involves processing a large dataset using Java. You need to utilize the Function interface and streams to efficiently manipulate the data and produce specific outputs. To ensure that your team has a thorough understanding of these concepts, you decide to create a challenging programming question that tests their knowledge and problem-solving skills.

For this assignment, consider a scenario where you have a dataset containing a list of employees in a company. Each employee object has attributes such as name, age, department, and salary.

For this assignment design a Java program that uses the Function interface and streams to perform the following operations:

Develop a program that reads the dataset and stores it in a collection.

Write a function using the Function interface that takes an employee object as input and returns the employee's name and department as a concatenated string.

Using streams, generate a new collection that contains the concatenated strings obtained from the previous step.

Enhance your program to find the average salary of all employees using the stream's built-in functions.

Generalize your program by incorporating a filter function that only includes employees whose age is above a certain threshold (e.g., 30 years).

Summarize the program by explaining the purpose, characteristics, and usage of the Function interface in Java, highlighting how it represents a function that takes an input and produces an output.

Compose a complete Java program that implements the given scenario, demonstrating your understanding of the Function interface, streams, and their practical applications in data manipulation.

You will be accessed based on the following criteria:

Program Structure and Correctness: The program correctly reads and stores the dataset, uses the Function interface to concatenate employee names and departments, utilizes streams to generate a new collection, calculates the average salary, and incorporates a filter function based on age threshold, producing accurate results for all operations.

Proper Usage of Function Interface and Streams: The program demonstrates a clear understanding of the Function interface by correctly implementing it to transform employee objects into concatenated strings, and effectively utilizes streams to process the dataset, perform operations, and chain multiple stream operations.

Efficiency and Performance: The program efficiently utilizes streams to process the dataset, minimizes memory usage and processing time, and takes advantage of stream's lazy evaluation and short-circuiting behavior to optimize performance.

Code Readability and Documentation: The code is well-structured, properly indented, uses meaningful variable and method names, includes comments to explain important sections, and provides a clear and concise explanation of the purpose and usage of the code components.

Additional Features: The program includes bonus functionalities or improvements that go beyond the specified requirements, showcasing creativity and innovation in enhancing the functionality or performance of the employee management system.

Remember to use appropriate variable names and follow coding best practices.

Submit the assignment in MS Word or PDF file. Your submission should contain the program code, explanation, and output screenshot

Grading criteria

Rubric

Program Structure and Correctness

Appropriate class structure, including necessary imports and packages are missing.

0 points

The program has appropriate class structure, including necessary imports and packages, reads the dataset and store it in a suitable collection, uses the Function interface to create a function that concatenates the employee's name and department, uses streams to generate a new collection with the concatenated strings. However, it does not calculate the average salary of all employees using stream's built-in functions and does not incorporate a filter function to include only employees above a certain age threshold. Also, there is no Functional interface to create a function that concatenates the employee's name and department, and there is no streams to generate a new collection with the concatenated strings.

15 points

The program has appropriate class structure, including necessary imports and packages, reads the dataset and store it in a suitable collection, uses the Function interface to create a function that concatenates the employee's name and department, uses streams to generate a new collection with the concatenated strings. However, it does not calculate the average salary of all employees using stream's built-in functions and does not incorporate a filter function to include only employees above a certain age threshold.

24 points

The program has appropriate class structure, including necessary imports and packages, reads the dataset and store it in a suitable collection, uses the Function interface to create a function that concatenates the employee's name and department, uses streams to generate a new collection with the concatenated strings, calculates the average salary of all employees using stream's built-in functions, incorporates a filter function to include only employees above a certain age threshold.

30 points

### Proper Usage of Function Interface and Streams

Function interface and its usage are missing.

0 points

Demonstrates a clear understanding of the Function interface and its usage. However, does not implement the ability to chain multiple stream operations effectively. Streams are not used to process the dataset and does not perform the required operations.

15 points

Demonstrates a clear understanding of the Function interface and its usage. Demonstrates the ability to chain multiple stream operations effectively. However, Streams are not used to process the dataset and does not perform the required operations.

24 points

Demonstrates a clear understanding of the Function interface and its usage. Demonstrates the ability to chain multiple stream operations effectively. Streams are properly used to process the dataset and perform the required operations.

30 points

### Efficiency and Performance

Streams are not used to process the dataset.

0 points

Uses streams efficiently to process the dataset, avoiding unnecessary iterations. However, it does not demonstrate an understanding of stream's lazy evaluation and short-circuiting behavior. Also, it does not minimize memory usage and processing time.

10 points

Uses streams efficiently to process the dataset, avoiding unnecessary iterations. Demonstrates an understanding of stream's lazy evaluation and short-circuiting behavior. However, it does not minimize memory usage and processing time.

16 points

Uses streams efficiently to process the dataset, avoiding unnecessary iterations. Demonstrates an understanding of stream's lazy evaluation and short-circuiting behavior, minimizes memory usage and processing time.

20 points

### Code Readability and Documentation

Code unstructured and not properly indented.

0 points

Code well-structured and properly indented. However, variable and method names do not congruent and do not follow standard naming conventions. Also, there is no comment explaining the purpose of important sections or complex operations.

5 points

Code well-structured and properly indented, Variable and method names meaningful and follows standard naming conventions. However, there is no comment explaining the purpose of important sections or complex operations.

8 points

Code well-structured and properly indented, Variable and method names meaningful and follows standard naming conventions, code includes comments to explain the purpose of important sections or complex operations.

10 points

### Additional Features

The program did not implement additional features or improvements beyond the requirements stated in the question.

0 points

The program implements additional features or improvements beyond the requirements stated in the question. However, there is no creative solution enhancing the program's functionality and performance.

5 points

The program implements additional features or improvements beyond the requirements stated in the question, creative solutions enhancing the program's functionality. However, there is no creative solution enhancing the program’s performance.

8 points

The program implements additional features or improvements beyond the requirements stated in the question, creative solutions enhancing the program's performance and functionality.

10 points