CO3 Assignment

1. In a retail analytics system, you have a list of sales transactions with attributes like product, amount, and category. Use the Stream API to filter transactions above $100, map to extract amounts, and reduce to calculate the total sum. Write the code and explain each operation (filter, map, reduce). Extend it to collect results into a List and discuss how streams improve code readability over traditional loops.
2. In an HR system, employees have attributes like name, salary, and join date. Implement the Comparable interface for an Employee class to sort by name naturally. Then, create a custom Comparator to sort by salary descending. Write code to sort an ArrayList using both approaches and explain the differences between Comparator and Comparable, including when to use each for custom sorting.
3. Design a simple chat application where messages are queued for delivery and user IDs map to their message histories. Use a HashMap> to store user-message pairs and a Queue (implemented with LinkedList) for pending messages. Write code to enqueue messages, dequeue them for processing, and retrieve a user's history from the map. Discuss the time complexity of key operations in HashMap and why Queue is appropriate for FIFO processing.
4. In a GUI application, you handle button clicks and data filtering using lambdas. Use a Predicate lambda to filter strings longer than 5 characters from a list. Then, apply a Consumer lambda to print each filtered item. Write the code integrating these functional interfaces and explain how lambda expressions simplify code compared to anonymous classes, including their role in functional programming.
5. Consider an online shopping platform where orders can fail due to insufficient stock or invalid payment details. Create a custom exception class OrderProcessingException that extends a checked exception. Demonstrate its usage in a method that processes an order, using throw to raise the exception when conditions are not met. Include try-catch blocks to handle it and explain the difference between checked and unchecked exceptions in this context.
6. In a project management app, tasks need to be stored and manipulated efficiently. Compare ArrayList and LinkedList in terms of performance for adding/removing elements. Write code to use a LinkedList for tasks that are frequently inserted/deleted, and convert it to a HashSet to remove duplicates. Explain why HashSet is suitable here and demonstrate adding elements while handling potential null values.