HANDS-ON

Hands-on No. : **7**

Topic : **Stream API Basics**

Date : **12.08.2025**

Solve the following problems

Question No.	Question Detail
1	Write a program that uses a stream to generate an infinite sequence of natural numbers starting from 1. • Filter numbers that are divisible by 5. • Limit the stream to the first 5 such numbers. • Print them using forEach().
2	• Given a list of integers, create a new list where each integer is squared. • Given a list of strings, filter those that start with the letter "S". • Sort a list of strings alphabetically using streams. • Convert a list of integers to a **Set** of squared values (to remove duplicates). • Use forEach() to print each squared value.
	You are tasked with developing a Java application to manage and analyze product information for an e-commerce system. The product data is stored as an ArrayList<Product>, where You are working as a Java developer for a retail inventory management system. Your task is to model product data using object-oriented principles and perform various data processing operations using Java's **Stream API**. The system maintains a list of products, each with details such as name, category, brand, price, stock quantity, units sold, and creation date. You are required to: 1. **Model the product** using a Product class with appropriate fields and methods. 2. **Store a list of products** in memory using an ArrayList<Product>, where each Product object contains the following fields: • id: Product ID (integer) • name: Product name (string)

It is going to be hard but, hard does not mean impossible.

	• category: Product category (e.g., Laptop, Mobile, Headphones) • brand: Manufacturer brand (string) • price: Product price (double) • quantity: Units available in stock (integer) • unitsSold: Total units sold so far (integer) • createdDate: Date the product was added (LocalDate) Consider the sample data below for the processing, new Product(1, "iPhone 13", "Mobile", "Apple", 69999.00, 20, 150, LocalDate.of(2024, 5, 1)), new Product(2, "Galaxy S22", "Mobile", "Samsung", 64999.00, 15, 120, LocalDate.of(2024, 6, 15)), new Product(3, "Dell Inspiron 15", "Laptop", "Dell", 55999.00, 10, 70, LocalDate.of(2024, 4, 10)), new Product(4, "MacBook Air M2", "Laptop", "Apple", 99999.00, 5, 90, LocalDate.of(2024, 7, 1)), new Product(5, "Sony WH-1000XM4", "Headphones", "Sony", 19999.00, 30, 200, LocalDate.of(2024, 3, 20)), new Product(6, "HP Pavilion x360", "Laptop", "HP", 49999.00, 12, 40, LocalDate.of(2024, 5, 10)), new Product(7, "OnePlus Nord CE", "Mobile", "OnePlus", 24999.00, 25, 100, LocalDate.of(2024, 4, 5)), new Product(8, "Lenovo Tab M10", "Tablet", "Lenovo", 17999.00, 18, 60, LocalDate.of(2024, 6, 1)), new Product(9, "Samsung Galaxy Tab A7", "Tablet", "Samsung", 20999.00, 20, 80, LocalDate.of(2024, 6, 20)), new Product(10, "Realme Buds Air 3", "Headphones", "Realme", 3999.00, 50, 250, LocalDate.of(2024, 2, 28)) Execute the operations below using stream API
4	1. **Filter High-Priced Products** Retrieve all products that are priced above ₹50,000. 2. **Sort by Price (Descending Order)** Sort the products based on price from highest to lowest. 3. **Find the Costliest Product** Identify and display the product with the maximum price using a terminal operation.

	4. **Calculate Total Inventory Worth** Compute the overall inventory value by multiplying each product's price with its quantity and summing the result. 5. **Group Products by Category** Organize products into groups based on their category and display the number of products in each group. 6. **Map Brand to Product Names** Create a mapping where each brand is associated with a list of its product names. 7. **Detect Low Stock Items** Check if any product has a quantity less than 5 in stock. 8. **Compute Average Product Price** Calculate and display the average price across all products. 9. **Identify Top 3 Best-Sellers** Retrieve the top three products with the highest number of units sold. 10. **Fetch Recently Added Products** List all products added in the last 60 days, based on their creation date.
5	11. **Find Products by Category and Price Range** Retrieve all products that belong to the "Laptop" category and are priced between ₹40,000 and ₹80,000. 12. **Find Products with Zero Sales** List all products that have unitsSold equal to 0. 13. **Count Products per Brand** Count how many products each brand offers and display the result as a map. 14. **Find the Product with the Highest Stock Value** Determine which product has the highest stock value (i.e., price × quantity). 15. **List Unique Product Categories** Extract and display a distinct list of all product categories. 16. **Check If All Products Are In Stock** Verify whether all products have at least one unit in stock (i.e., quantity > 0). 17. **Generate Summary Statistics for Product Prices** Use DoubleSummaryStatistics to display the count, min, max, average, and sum of product prices.

	18. **Sort Products Alphabetically by Name** Display all products sorted in ascending order of their names. 19. **Group Products by Brand and Count Units Sold** Group products by brand and calculate the total units sold for each brand. 20. **Find the Most Recently Added Product** Identify and display the product with the most recent createdDate.