

Faculty of Engineering, University of Jaffna,

Department of Computer Engineering.

EC5080: Software Construction

Lab 04

Duration: 3 Hours

Lecturer: Ms. Sujanthika M

Chapter: Collections

Objectives:

- Apply the Java Collections Framework effectively to model real-world problems
- Use Sets and Maps to maintain uniqueness and lookup relationships
- Apply Iterator and ListIterator safely for modifications

Task 1: Product and Sale Entry Structures

1. Create a class SaleEntry to represent a sale:
 - a. String productId, int quantity, double pricePerUnit
 - b. Override equals() and hashCode() for uniqueness based on productId
2. Create a class Product:
 - a. String productId, String name
 - b. List<SaleEntry> sales
3. Create at least 5 products, each with 3 or more sales.
4. Print each product's total revenue (sum of quantity * pricePerUnit)
5. Prevent duplicate SaleEntry using a Set<SaleEntry> instead of List and show how it helps.

Task 2: Ranking Products by Revenue

1. Implement Comparable<Product> in the Product class to sort by total revenue.
2. Store all products in a TreeSet<Product> and print them in descending order of revenue.
3. Use a HashMap<String, Product> to allow quick lookup by product ID.
4. Implement a method to:
 - a. Find the top 3 revenue-generating products
 - b. Print their names, IDs, and revenue

Task 3: Removing Inactive Products Using Iterators

1. A product is "inactive" if:
 - a. It has no sales, or
 - b. Its total revenue is below 100
2. Remove such products from the original list using:
 - a. An Iterator
 - b. A ListIterator
3. Try the same with a for-each loop and record the behavior.

Write a brief explanation about:

- Why modification during for-each loop fails
- How iterator safely allows removal

Task 4: Performance Comparison Across Collections

1. Simulate 10,000 products with random sale data.
2. Store them in:
 - a. `ArrayList<Product>`
 - b. `HashSet<String>` for product IDs
 - c. `TreeMap<String, List<SaleEntry>>` for mapping product name → sales
3. Measure time to:
 - a. Insert all products
 - b. Search 100 random product IDs
 - c. Remove 100 random products

Present results in this table:

Collection	Insert Time	Search Time	Remove time
ArrayList			
HashSet			
TreeMap			

Task 5: Product Directory using HashMap and TreeMap

(20 marks)

1. Use a `HashMap<String, String>` to store a product directory (product name → supplier contact).
2. Add 5 entries and demonstrate:
 - a. `putIfAbsent()`
 - b. `getOrDefault()`
 - c. `containsKey()` and `containsValue()`
3. Convert it into a `TreeMap<String, String>` and print all suppliers in sorted order by product name.
4. Implement a method to print all suppliers whose names start with a given letter.

Marking Scheme:

Task	Details	Marks
Task 1	Tuple class, revenue calculation, uniqueness logic	20
Task 2	TreeSet usage, Comparable logic, lookup	20
Task 3	Safe removal with iterators and explanation	10
Task 4	Timing code, table, and justification of findings	20
Task 5	Map operations, TreeMap sort, search	20
Code quality	Clear structure, naming, modular design	10