

## Advanced Programming (OOP) Lab

### Assignment–7

#### Report Instructions

This document specifies the mandatory structure and assignment-specific questionnaires for the handwritten report corresponding to Assignment–7: Collection Framework in Java. The report will be evaluated for understanding of collection semantics, choice of data structures, and reasoning about performance, ordering, and uniqueness.

#### General Submission Rules

- The report must be handwritten and submitted individually.
- Do not rewrite complete programs; include only short, relevant code fragments.
- Explanations must justify data structure choices and operational behavior.
- All implemented requirements and attempted exercises or bonus tasks must be covered.
- Superficial descriptions of collections without reasoning will be penalized.

***Tip:** Use multiple colors for highlighting your explanations, references and annotations.*

#### Section A: Collection Selection Rationale

Explain why ArrayList, HashMap, and HashSet are used in this assignment. For each collection, state the problem it solves and the property (ordering, uniqueness, key–value access) that makes it suitable.

#### Section B: List-Based Data Handling

For the list of books:

1. Explain how objects are stored and accessed in an ArrayList.
2. Describe how sorting is achieved using Comparator.
3. Explain what happens internally when the list grows beyond its initial capacity.

#### Section C: Map Usage and Key Semantics

For the student marks stored in a map:

- Explain how keys and values are managed internally
- Justify the choice of HashMap over other Map implementations
- Describe what happens when a duplicate key is inserted

#### Section D: Set Semantics and Uniqueness

Explain how HashSet ensures uniqueness of elements. If TreeSet or LinkedHashSet is used, explain how ordering differs. Describe the role of hashCode() and equals() in maintaining set behavior.

## Section E: Traversal Techniques

Explain the difference between traversal using for-each loops and Iterators. Describe one situation where an Iterator is mandatory. Include a small code fragment to illustrate safe traversal.

## Section F: What-If and Performance Analysis

Answer any two of the following:

1. What changes if ArrayList is replaced with LinkedList?
2. How does performance change if TreeMap is used instead of HashMap?
3. What issues arise if mutable objects are used as keys in a Map?
4. What breaks if equals() is overridden but hashCode() is not?

## Section G: Bonus or Extension Explanation

If you implemented any bonus tasks (LinkedHashMap, serialization, custom exception):

- Explain the design intent
- Describe how correctness is enforced
- Explain how the extension integrates with existing collections

## Section H: Individual Variation Task

Describe one individual modification or enhancement, such as:

- Alternate sorting criteria
- Additional search functionality
- Custom validation logic

Explain its impact on collection behavior.

## Section I: Debugging and Learning Reflection

Describe one non-trivial issue encountered while working with collections. Explain how it was diagnosed and what deeper understanding of the Collection Framework was gained.

## Section J: AI Usage Disclosure

Declare whether AI or online tools were used. Specify:

- Which collection-related logic was assisted
- How the solution was validated and adapted