CSC 212 Project

E-Commerce Inventory & Order Management System

Fall 2025

Phase II – Due Date November 30th, 2025

Objective

Phase II builds upon your Phase I project. In this phase, the focus shifts from linear data structures to **logarithmic-time data structures** such as **Binary Search Trees (BSTs)** or **AVL Trees (optional)** to improve search efficiency, data organization, and support for complex queries.

The implementation will be refactored to:

1. Replace linear searches with logarithmic searches using BSTs/AVLs.
2. Support **range-based queries** and **sorted traversals**.
3. Introduce **performance evaluation** comparing linear (Phase I) and logarithmic (Phase II) search complexities.

Data Structures Upgrade

- Products, Customers, and Orders must now be stored using a **BST or AVL Tree** (student's choice).
- Each node must store key-value pairs:
  - Product keyed by productId.
  - Customer keyed by Customer Name.
  - Order keyed by orderId.

Requirements

- Use same data set sent in Phase I
- **Insert / Update Product:** must operate in **O(log n)**.
- **Search Product by ID:** must use logarithmic search
- **Range Query by Price:** return all products whose price falls within a given range [minPrice, maxPrice].
- **Insert / Search Customer**: via BST/AVL (keyed by customerId).
- **Customer Order History**: Retrieve all orders for a given customer efficiently.

Advanced Query Requirements

Each query must be optimized using BST/AVL traversal logic:

1. **Find All Orders Between Two Dates** (use in-order traversal).
2. **List All Products Within a Price Range.**
3. **Show the Top 3 Most Reviewed or Highest Rated Products.**
4. **List All Customers Sorted Alphabetically.**
5. **Given a Product ID, Display All Customers Who Reviewed It (sorted by rating or customer ID).**

. **Deliverables**

- A complete class diagram that shows classes, methods, relationships
- Your new codebase for this phase which should include all of your code
- A written report that shows all classes, methods, also the analysis of time complexity and space complexity.
- Big-O analysis comparison between Phases I and II.
- All submissions will be using a single zip folder through LMS
- One mark bonus for using AVL Trees

Rules

- All data structures used in this assignment must be implemented by the student. The use of Java collections or any other data structures library is strictly forbidden.
- Posting the code of the assignment or a link to it on public servers, social platforms or any communication media including but not limited to Facebook, Twitter or WhatsApp will result in disciplinary measures against any involved parties.
- All submitted code will be automatically checked for similarity, and if plagiarism is confirmed penalties will apply.
- You may be selected to discuss your code with an examiner at the discretion of the teaching team. If the examiner concludes plagiarism has taken place, penalties will apply.
- You are allowed up to 3 team members per project.