## ArrayList (with Product Object, CRUD + Business Logic)

**Problem Statement:**  
Design a **Product Inventory Management System** using an ArrayList<Product> where each Product object contains attributes like id, name, price, and quantity.

Your task is to implement the following:

### 🔸 CRUD Operations

1. **Create** – Add new Product objects into the ArrayList.
2. **Read** – Display all products in the inventory.
3. **Update** – Update a product’s price or quantity by searching with the id.
4. **Delete** – Remove a product from the list by id.

### 🔸 Business Logic Method

* Write a method to **calculate the total inventory value**, i.e., sum of price × quantity for all products.

**HashMap (with Object)**

**Problem Statement:**  
Design a **Library Management System** using a HashMap<Integer, Book> where the key is the **Book ID** and the value is a Book object (with properties like title, author, price).

Your task is to:

1. Add multiple books using put().
2. Retrieve a book by its ID using get().
3. Check if a given Book ID exists using containsKey().
4. Check if a particular Book object exists using containsValue().
5. Display the total number of books using size().
6. Iterate through all entries using entrySet(), keySet(), and values().
7. Update book details (like price) using replace().
8. Remove a book by ID using remove().
9. Clear all books using clear().
10. Verify if the collection is empty using isEmpty().

**🔹 Stack (with Integer values)**

**Problem Statement:**  
Implement a stack to store a series of integer values representing **daily temperatures**. Perform the following:

1. Push 5 temperature values onto the stack.
2. Pop the top 2 values and display them.
3. Peek at the current top element without removing it.
4. Check if the stack is empty.

**🔹 Queue (with Integer values)**

**Problem Statement:**  
Simulate a **ticket booking system** where customers are represented by their token numbers (integers). Perform the following:

1. Enqueue 5 customer tokens into the queue.
2. Dequeue the first 2 customers and display them.
3. Peek at the next customer in line without removing.
4. Check if the queue is empty.

**🔹 LinkedList (with Integer values)**

**Problem Statement:**  
Create a **task manager** using a linked list where each integer represents a task priority. Perform the following:

1. Add 5 tasks (integers) into the list.
2. Insert a new task at the beginning and another at the end.
3. Remove a task from the middle.
4. Display the first and last task using getFirst() and getLast().
5. Check if a specific task priority exists in the list.