**Intern Exercise: Tirupati Darshan Booking System (Simple OOP Version)**

**Problem Statement**

You are tasked with building a **Tirupati temple darshan booking system**:

* Devotees can book **Standard** or **VIP** darshan.
* The system should:
  1. Maintain **all bookings** in order received.
  2. Track **unique devotees**.
  3. Assign **VIP devotees first**, then Standard devotees.
  4. Map devotees to their **darshan slot/time**.

**Collections to Use**

|  |  |  |
| --- | --- | --- |
| Feature | Collection | Purpose |
| All bookings | List<Booking> | Preserve order |
| Unique devotees | Set<Devotee> | No duplicates |
| Standard queue | Queue<Booking> | First-come-first-serve for Standard devotees |
| VIP queue | Queue<Booking> | Serve VIP devotees first |
| Mapping devotee → slot | Map<Devotee, String> | Track assigned slots |

**Step-by-Step Instructions for Interns**

**Step 1: Create Classes**

1. **Devotee Class**
   * Fields: name (String), id (int)
   * Methods: constructor, toString()
   * Override equals() and hashCode() for uniqueness in Set
2. **Booking Class**
   * Fields: Devotee devotee, type (String: VIP/Standard)
   * Methods: constructor, toString()
3. **DarshanManager Interface**
   * Methods: addBooking(), assignSlots(), displayBookings(), displayUniqueDevotees(), displayDarshanSlots()
4. **DarshanBookingSystem Class**
   * Implements DarshanManager
   * Holds all collections and implements the required methods

**Step 2: Logic Flow**

1. Take **number of devotees** as input.
2. For each devotee:
   * Input **name** and **VIP/Standard type**
   * Create Devotee and Booking objects
   * Add to **List**, **Set**, and **VIP/Standard queue**
3. Assign **darshan slots** sequentially (9:00, 9:15, 9:30…)
   * **VIP queue first**, then **Standard queue**
   * Map devotees to their assigned slot in Map
4. Display:
   * All bookings
   * Unique devotees
   * Assigned darshan slots

**✅ Learning Points**

1. Use **List** to maintain all bookings in order.
2. Use **Set** to maintain unique devotees.
3. Use **Queue** to manage VIP and Standard devotees separately.
4. Use **Map** to assign and look up darshan slots.
5. Use **OOP principles**: interface, classes, modular methods.
6. No lambdas, beginner-friendly for interns to understand.