**Hypothesis Brief**

After analyzing the tap log against the candidate rules, the following set explains the observed charges:

* **R1 – Base Fare Rule**: rs 25 is the starting charge for any first tap. Verified at taps like 07‑01 07:20 G → rs 25.
* **R2 – Peak Time Rule**: During 8–10 AM and 6–8 PM, fare increases by +50% (rs 37.5 from base). Seen at 07‑01 08:01 G and 07‑02 18:02 Y.
* **R3 – Transfer Window Rule**: A tap within 30 minutes of a paid tap is free. Example: 07‑01 08:30 R → rs0 after 07‑01 08:01 G (rs 37.5).
* **R4 – Night Discount**: From 10 PM–midnight, 20% discount on base → rs 20. Seen at 07‑01 22:15 Y (rs 20).
* **R5 – Post‑Midnight Discount**: From 12–4 AM, 35% discount → 25 × (1 – 0.35) = rs 16.25. Matches 07‑02 00:45 X.

**Conclusion:**  
All 5 rules R1–R5 explain the dataset, when applied sequentially in priority order:  
Base → Peak → Discounts (time-based) → Transfer Window.

**Class Design Note**

* **Tap**: Represents a single tap event (datetime, line, station). Immutable record.
* **FareRule** (abstract): Interface for all rules. Each rule implements apply(fare, tap, history).
* **BaseFareRule**: Sets default fare = ₹25.
* **PeakFareRule**: Checks time; applies +50% surcharge if in peak window.
* **TransferRule**: Checks if last paid tap within 30 mins; if yes → fare = 0.
* **NightDiscountRule**: Applies 20% discount if after 10 PM to midnight.
* **PostMidnightDiscountRule**: Applies 35% discount if between 12–4 AM.
* **TariffEngine**: Core engine; takes a list of rules + switch toggles. Iterates through rules in set order, applying transformations to compute final fare.
* Extension Points: Just add new FareRule subclasses, or flip booleans to turn them on/off.