Shopping Cart with Billing Engine
(C++ | No External Libraries | File Usage Allowed)

Problem Statement

Design and implement a Shopping Cart with Billing Engine using C++. The system should allow users to add products to a cart, apply discounts, calculate tax, and generate a final bill. The system should be executable and must perform real calculations and file operations.

Functional Requirements

1. Product Management

- Products must have a unique Product ID, name, and price (in INR).
- Product price must be non-negative.
- Product details must not change once created.

2. Cart Management

- Add products to the cart with a specified quantity.
- If the same product is added again, update the quantity.
- Remove products from the cart.
- Update quantity of existing products.
- Display all items in the cart.
- Calculate subtotal as price × quantity.

3. Discount Handling

3.1 Percentage Discount

- Apply a percentage-based discount on cart subtotal (e.g., 10% off).

3.2 Flat Discount

- Apply a fixed discount amount (e.g., ■500 off).

3.3 Threshold-Based Discount

- Apply a discount only if subtotal crosses a defined threshold.
- Example: ■1000 off if subtotal ≥ ■10,000.

Discount Rules

- Multiple discounts can be applied together.
- Discounts must be applied before tax calculation.
- Total discount must not exceed the cart subtotal.

- Final payable amount must never be negative.

4. Tax Calculation

- Tax must be applied after all discounts.
- Tax rate should be configurable (e.g., 18% GST).

5. Billing Engine

- Calculate subtotal, total discount, tax amount, and final payable amount.
- Billing logic must not modify cart data.
- Billing should be repeatable without changing cart state.

6. Invoice Generation

- Generate itemized invoice with product name, quantity, unit price, and total.
- Display subtotal, discount, tax, and final payable amount.
- Invoice must be printed on console and saved to invoice.txt.

7. File Handling

- Invoice must be written using standard C++ file handling.
- Invoice file should be overwritten on each generation.

8. User Interaction (CLI)

- Menu-driven interface for cart operations.
- Graceful handling of invalid input.

9. Constraints

- Only standard C++ libraries allowed.
- No external frameworks.
- System must be modular and extensible.
- Hardcoded outputs without logic are not allowed.

10. Assumptions

- Product data can be hardcoded.
- No database required.
- Single user and single cart per execution.

Sample Invoice Output (CLI Style)

```
----- INVOICE -----

Laptop     x1   ■50000
Mouse      x2   ■1000
-------------------
```

```
Subtotal: ■51000
Discount: ■5000
Tax (18%): ■8280

--------------------

Total Payable: ■54280
```

Evaluation Criteria

- Correctness of calculations.
- Handling of edge cases.
- Clean separation of responsibilities.
- Ease of adding new discount or tax rules.
- Proper use of file handling.