Perfect 👍 You’re now looking at a **specific WSTG area: Test Business Logic Data Validation (WSTG-BUSL-02)**.  
Here are precise notes for study 👇

**📝 Notes: Test Business Logic Data Validation**

**🎯 Purpose**

To verify if the application properly **validates user input against defined business rules**, beyond just technical input validation (like XSS/SQLi checks).

* Focus: **Does the app allow operations with values that should be disallowed according to business logic?**

**⚡ Common Examples of Business Logic Data Validation Flaws**

1. **Negative or Zero Values**
   * Ordering an item with price = -100 → system credits money to user.
2. **Exceeding Limits**
   * Transferring 9999999 when the limit is 5000.
3. **Invalid Combinations**
   * Buying out-of-stock items or applying multiple coupons not intended to stack.
4. **Date/Time Manipulation**
   * Submitting expired vouchers or backdating insurance claims.
5. **Boundary Testing**
   * Submitting 0, null, or extreme values to bypass checks.

**🔍 How to Test**

1. **Understand Business Rules**
   * Collect rules from documentation, user guides, or by observing workflows.
   * Example: "Coupon discount ≤ 50% of product price".
2. **Tamper with Data**
   * Use Burp Suite, Postman, or manual requests to modify values before submission.
   * Try negative numbers, huge values, empty fields, invalid states.
3. **Check Server-Side Validation**
   * Ensure validation is enforced on the server, not just client-side JavaScript.
4. **Look for Logical Inconsistencies**
   * Does the system allow actions outside the intended rules?

**🛡️ Mitigation**

* Enforce **strict server-side validation** against defined business rules.
* Apply **whitelisting for values and ranges** (not just blacklisting).
* Implement **input normalization** (no hidden bypasses).
* Add **business rule unit tests** during development.

✅ **Key Takeaway:**

This test checks if applications properly **enforce business constraints on data values**.  
Failure → attackers exploit logical loopholes (e.g., free orders, unlimited discounts, bypassing financial limits).

Do you want me to also prepare a **ready-to-use checklist of test cases** (like “try negative values, max integer, expired dates, etc.”) that you can directly apply while testing?