Assignment Title: Implementing the Biba Integrity Model in Java

---

Assignment Description:

The Biba Integrity Model is designed to maintain data integrity by preventing unauthorized modifications. In this assignment, you will implement a Java program to enforce the two core principles of the Biba model: the *simple integrity property* ("no read down") and the *integrity *-property* ("no write up"). Your program will simulate a system with users, objects, and integrity levels, enforcing Biba's rules for read and write operations.

Your program will:

1. Define users and objects with different integrity levels.
2. Enforce Biba's rules for access requests.
3. Provide an interactive interface for simulating read and write operations.

Requirements:

1. **System Components**:
    - Define a set of integrity levels (e.g., Low, Medium, High, Very High).
    - Define users and objects, each assigned a specific integrity level.
2. **Access Control Rules**:
    - Enforce the *simple integrity property*: users cannot read objects with a lower integrity level.
    - Enforce the *integrity *-property*: users cannot write to objects with a higher integrity level.
3. **Input and Output**:
    - Allow the user to simulate operations by specifying:
        - User ID
        - Object ID
        - Operation (read/write)
    - Display whether the operation is permitted or denied based on Biba's rules.

4. **Error Handling**:
    - Handle invalid inputs (e.g., non-existent users or objects).
    - Provide meaningful error messages for denied access.
5. **Documentation and Testing**:
    - Include comments explaining each part of the code.
    - Provide test cases to demonstrate the enforcement of Biba rules.

Deliverables:

1. **Java Source Code**:
    - Submit the .java file(s), ensuring the code is well-documented and easy to follow.
2. **Test Results**:
    - Provide a document summarizing test cases, including:
        - User and object integrity levels.
        - Attempted operations (read/write).
        - Whether the operations were allowed or denied.
3. **Readme**:
    - Include a README file in Word or PDF format explaining how to run your program and any dependencies.
4. **Optional Enhancements** (Extra Credit):
    - Implement a feature for dynamic integrity level changes for users and objects.
    - Add a logging system to track all access requests and their results.

Submission Guidelines:

1. Submit your .java file(s) and test results via Blackboard.

2. Include a README file in Word or PDF format explaining how to run your program and any dependencies.

10-Point Rubric:

Criteria	Points	Description
Correct Enforcement of Simple Integrity Property	2	Accurately prevents users from reading objects with lower integrity levels.
Correct Enforcement of *Integrity - Property	1	Accurately prevents users from writing to objects with higher integrity levels.
Integrity Level Implementation	1	Properly defines and applies integrity levels to users and objects.
User Input Handling	1	Handles user inputs for IDs and operations with appropriate validations.
Output Clarity	1	Clearly indicates whether operations are allowed or denied and provides reasons.
Error Handling	1	Provides meaningful error messages for invalid inputs or denied operations.
Documentation/Comments	1	Includes clear comments explaining major code sections and program logic.
Test Cases and Results	1	Provides at least three test cases demonstrating the correct enforcement of Biba rules.
Optional Enhancements	1	Implements dynamic integrity level changes or a logging system (extra credit).
Overall Functionality	1	Program runs correctly and enforces Biba rules as specified.