Republic of the Philippines
Laguna State Polytechnic University
Province of Laguna

Machine Problem No. 1			
Topic:	**Propositional Logic**	**Week No.**	2-3
Course Code:	**CSST101**	**Term:**	1st Semester
Course Title:	**Advance Knowledge Representation and Reasoning**	**Academic Year:**	2025-2026
Student Name	Capili, Judeelyn M.	**Section**	
Due date		**Points**	

University Logic Rules Application

I. Title

Assessment Task: Applying Propositional Logic in Real-World Scenarios through a Mini Expert System

II. Intended Learning Outcomes (ILOs)

At the end of this activity, students should be able to:

1. Translate real-world conditions into propositional logic expressions.
2. Apply logical implication (P → Q) to decision-making scenarios.
3. Develop a Python program that implements logic rules.
4. Record and analyze system results using CSV as a simple database.
5. Communicate findings through a short written report.

III. Instructions for Students

1. **Download or recreate the Mini Expert System** provided in class.
2. Run the program and test at least **3 different students** with different conditions for:
    - Attendance Rule ○ Grading Rule ○ Login System Rule
    - Bonus Points Rule
3. Verify that all results are logged in the **CSV file (logic_results.csv)**.
4. Extend the program by **adding one new rule of your own**. Examples:
    - Library borrowing (If ID is valid → Allowed to borrow books). ○ Enrollment clearance (If fees are paid → Enrollment confirmed).
    - Laboratory access (If safety gear is worn → Access granted).

5. Submit the following:
   - Source code (.py file). ○ Generated **CSV file** with results from at least 3 students.
   - A **short report (1–2 pages)** containing:
     - Explanation of rules tested.
     - Screenshots of program runs.
     - Description of the new rule you added.

IV. Assessment Criteria (Rubric)

Criteria	Excellent (100%)	Proficient (85%)	Developing (70%)	Beginning (50%)
Logic Translation	All rules correctly translated into propositional logic.	Minor errors in translation.	Some rules incorrectly expressed.	Little/no understanding of logic.
Program Implementation	Program runs smoothly with all rules implemented correctly.	Program runs with minor errors.	Program partially working.	Program does not run.
Extension Rule	Original and correctly implemented additional rule.	Additional rule included but minor errors.	Additional rule unclear/incomplete.	No additional rule added.
CSV Logging & Data Handling	All results properly logged and organized in CSV.	Most results logged correctly.	CSV incomplete or inconsistent.	No CSV output.
Report & Presentation	Clear, well-organized, with screenshots and explanations.	Clear but missing details.	Report incomplete or unclear.	No report submitted.

V. Sample Assessment Questions (for Written Part)

1. Represent the following in propositional logic:
   a. *"If a student is late, then they must bring an excuse letter."*
   b. *"If a grade ≥ 75, then the student passes."*
2. Why is the rule **P → Q** considered satisfied when **P is false**?
3. Examine the following CSV excerpt:

```
2025-09-04 14:33:15, Juan Dela Cruz, Grading Rule, Satisfied ✅
2025-09-04 14:34:22, Ana Santos, Login Rule, Access denied ❌
```

- What does each row represent?
- Which logical implication failed and why?

I. Program Overview

The University Logic Rules System is a Python-based mini expert system that applies the concept of propositional logic—specifically the rule of implication (P → Q)—to common university scenarios. The program evaluates whether logical conditions are satisfied or violated based on user input and logs all outcomes into a CSV file for analysis. Each rule represents a real-life policy such as attendance requirements, grading standards, login verification, and eligibility for bonuses. Through this system, logical reasoning is transformed into executable code, demonstrating how AI concepts can support automated decision-making and policy checking in a university context.

II. Rules Implemented

The system implements several logical rules that follow the implication structure (P → Q).

The Attendance Rule states that if a student is late (P), then they must bring an excuse letter (Q). This ensures that tardiness is justified, aligning with school discipline standards.

The Grading Rule follows the logic if a student's grade is 75 or higher (P), then the student passes (Q). This checks if the grading system adheres to the standard passing threshold.

The Login System Rule represents the statement if the entered password is correct (P), then access is granted (Q). This simulates a simple authentication process where access depends on correct credentials.

The Bonus Points Rule uses the statement if a student has regular attendance (P), then the student is eligible for bonus points (Q). It reflects reward systems in classrooms that encourage consistent attendance.

Lastly, the Library Borrowing Rule, which was added as the required extension, follows the proposition if a student's ID is valid (P), then they are allowed to borrow books (Q). This ensures that only verified students can access library materials.

Each of these rules operates under the same principle of logical implication, determining whether the given conditions lead to valid conclusions based on truth values.

III. Sample Program Runs

To test the system, several student scenarios were simulated. For instance, Ana Reyes was marked as late but brought an excuse letter, resulting in a satisfied implication. Mark Lopez entered a grade of 82, meeting the grading rule's condition. Ella Cruz entered an incorrect password; however, since the premise (P) was false, the implication remained true, showing "Access granted ✓" due to the nature of logical implication. Other tests confirmed that bonus eligibility and library borrowing conditions were properly evaluated. These examples illustrate how the program translates human decision-making into logical computation, providing consistent and objective results.

IV. CSV Logging Explanation

The system automatically records each result in a file named logic_results.csv, which serves as a simple database. Every time a student is evaluated under a rule, the program logs the timestamp, student name, rule title, and outcome. Each row represents one test instance, allowing instructors to review all evaluations later. For example, if a student was late but did not bring an excuse letter, that entry would be marked as "Violated ✗" in the CSV. This data logging process not only tracks student performance but also demonstrates how logic-based systems can produce audit trails for verification and transparency.

V. Answers to Written Questions

1. Represent the following in propositional logic:
a. The statement "If a student is late, then they must bring an excuse letter" can be expressed as P → Q, where P represents "The student is late" and Q represents "The student brings an excuse letter." This logical form means that the truth of being late implies the necessity of having an excuse letter.

b. The statement "If a grade is greater than or equal to 75, then the student passes" can also be expressed as P → Q, where P is "The student's grade is ≥ 75" and Q is "The student passes." This expresses the passing condition based on academic performance. Both statements follow the same structure of implication, which is central to propositional logic.

2. Why is the rule P → Q considered satisfied when P is false?
In propositional logic, an implication (P → Q) is defined as true in all cases except when P is true and Q is false. This means that if the premise (P) is false, the implication is automatically considered true—a concept known as vacuous truth. For example, if a student is not late (P is false), then the rule "If late, must bring an excuse letter" is not violated, even if the student has no excuse letter. The rule still holds because the condition requiring justification was never triggered. This principle highlights how logic handles situations where conditions do not apply, ensuring consistent reasoning in all possible truth values.

3. CSV Excerpt Analysis
Each row in the CSV file represents a single logical evaluation for one student under one rule at a specific time. The columns include the timestamp of the test, the student's name, the rule that was checked, and the outcome, which may be "Satisfied ✓" or "Violated ✗." When a rule is violated, it indicates that the antecedent (P) was true but the consequent (Q) was false—for example, the student was late but failed to bring an excuse letter. Conversely, if P is false or both P and Q are true, the implication is satisfied. This structure makes the CSV not only a data log but also an analytical record of how logical implications hold across different cases.

VI. Reflection and Learning Insights

Through this machine problem, I learned how abstract logical concepts such as implication (P → Q) can be represented and tested in real-world programming scenarios. The exercise helped bridge the gap between theoretical logic and practical implementation, showing how decision-making systems rely on structured reasoning. Creating and extending the rules also encouraged analytical thinking and attention to logical consistency. Additionally, using a CSV file for output demonstrated a simple yet effective way to record and organize system data. Overall, this activity deepened my understanding of how knowledge representation and propositional logic form the foundation for more advanced artificial intelligence systems.