Republic of the Philippines
Laguna State Polytechnic University
Province of Laguna

Machine Problem No. 2			
Topic:	**Topic 1.2: Logic-Based Representation**	**Week No.**	3-4
Course Code:	**CSST101**	**Term:**	1st Semester
Course Title:	**Advance Representation and Reasoning**	**Academic Year:**	2024-2025
Student Name		**Section**	
Due date	**September 02, 2024**	**Points**	

Machine Problem: Implementing a Logic-Based Model in Python

Objective:

Implement propositional and predicate logic in Python to understand the fundamental operations of logic-based representation and its application in AI.

Task Instructions:

1. **Propositional Logic Operations:**
   - **Write Python Functions:** Implement basic propositional logic operations including:
     - **AND (∧):** Logical conjunction.
     - **OR (∨):** Logical disjunction.
     - **NOT (¬):** Logical negation.
     - **IMPLIES (→):** Logical implication.
   - **Example:** Create functions such as and_operation(p, q), or_operation(p, q), not_operation(p), and implies_operation(p, q).

2. **Evaluate Logical Statements:**
   - **Create an Evaluation Function:** Develop a function that takes logical statements as input and evaluates their truth value.
   - **Example:** A function evaluate(statement, values) where statement is a logical expression and values is a dictionary mapping propositions to their truth values.

3. **Extend to Predicate Logic:**
   - **Support for Quantifiers:** Extend the logic functions to support predicate logic with quantifiers:
     - **FOR ALL (∀):** Universal quantifier.
     - **EXISTS (∃):** Existential quantifier.

- **Example:** Create functions forall(predicate, domain) and exists(predicate, domain) to evaluate predicate logic statements over a given domain.

4. **AI Agent Development:**
    - **Create a Simple AI Agent:** Develop an AI agent that uses logic to make decisions in a defined scenario. This could involve:
        - **Defining the Scenario:** For example, a simple decision-making task such as determining the best move in a game or choosing an action based on environmental conditions.
        - **Implementing the Decision Logic:** Use the logic functions to guide the AI agent's decisions.

Submission:

- **Python Scripts:**
    - Submit the Python scripts that implement the logic functions and AI agent.
    - Ensure your code is well-documented with comments explaining each function and its purpose.
- **Google Collab:**
    - Create a Google Collab Notebook that includes:
        - **Examples:** Demonstrate the use of each logic function with examples.
        - **Explanations:** Provide detailed explanations of how the logic operations work, including how the AI agent makes decisions.
        - **Results:** Show the outcomes of the AI agent's decisions in various scenarios.
- **Submission Format:**
    - Upload your Python scripts and Notebook to your GitHub repository.
    - Ensure the repository is well-organized, with folders and files clearly labeled (e.g., scripts/, notebooks/, README.md).

Grading Criteria:

Criteria	Excellent (10 points)	Good (8 points)	Fair (5 points)	Poor (2 points)
Implementation of Logic Functions	Accurate and efficient implementation of all required logic functions.	Mostly accurate with minor issues.	Basic implementation with significant errors.	Incomplete or incorrect implementation.
Extension to Predicate Logic	Correct and effective implementation of quantifiers in predicate logic.	Mostly correct with minor issues.	Basic implementation with significant errors.	Incorrect or incomplete implementation.
AI Agent Functionality	AI agent makes logical and correct decisions based on the implemented logic.	AI agent works with minor issues.	AI agent functions with significant errors.	AI agent fails to make correct decisions.
Jupyter Notebook Quality	Comprehensive, clear, and well-documented examples and explanations.	Adequate documentation with minor gaps.	Limited documentation with significant gaps.	Poor or incomplete documentation.
Code and Repository Organization	Well-structured, readable code and organized repository.	Mostly organized with minor issues.	Somewhat organized but lacks clarity.	Disorganized or unclear repository.