MASINDE MULIRO UNIVERSITY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE
BCS 227 LOGIC PROGRAMMING [21-MARCH-2024]

a. Explain how ontologies contribute to knowledge representation and the challenges they address. **[2 marks]**

b. In a game, it is known that "If I win, then I'll be happy." If you are happy, what can you infer about whether you won or not using propositional logic? **[4 marks]**

c. In Prolog, loop-like behavior is achieved through recursion rather than explicit loop constructs like in imperative languages. Explain this using an appropriate predicate, showing both tail recursion and non-tail recursion. **[6 marks]**

d. Imagine designing a rule-based expert system for diagnosing medical conditions. Discuss how logic programming, specifically Prolog, can be employed to represent rules and infer possible diagnoses based on input symptoms. What challenges might arise in this scenario? **[6 marks]**

e. Design a Prolog predicate named **average_grade** that takes a database of students with their grades and calculates the average grade of all students. The database should be represented as a list of facts where each fact represents a student and their grade, e.g., **student_grade(Student, Grade)**. Utilize built-in predicates such as **findall/3** for database manipulation to collect all grades and **sum_list/2** for arithmetic operations to calculate the total sum of grades. The predicate should be tested with a variety of test cases including an empty database, a database with multiple students, and edge cases such as when all students have the same grade. **[7 marks]**

f. Write a prolog database of facts and rules that automates courses on offer, lecturer(s) teaching each course(s), level at which the course is offered, lecture hall and lecture time for each course. Utilize the concepts of list, cut function or predicate, read() and write() functions. Include dummy data and relevant goals that simulate the outputs. **[7 marks]**

g. Design a Prolog program that monitors various aspects of a vehicle's condition. The program should include checks for the fuel level, air condition status, engine temperature, and vehicle capacity. Each aspect should be compared against specific limits, and appropriate signals should be generated based on the conditions. Ensure that the program checks the following conditions: The minimum acceptable fuel level should be set to 4 liters and if the fuel level falls below the minimum limit, the program should generate a signal indicating "Low fuel level". The expected status of the air condition should be set to "on" and if the air condition status is not "on", the program should generate a signal indicating "Air condition is not on". The maximum safe engine temperature should be set to 90 degrees Celsius and if the engine temperature exceeds the maximum limit, the program should generate a signal indicating "Engine temperature is too high". The maximum capacity of the vehicle should be set to 5 passengers and if the number of passengers exceeds the maximum limit, the program should generate a signal indicating "Vehicle capacity exceeded". Your Prolog program should allow input parameters for the current fuel level, air condition status, engine temperature, and vehicle capacity for testing purpose. It should then output a list of signals indicating any issues detected with the vehicle's condition. **[8 marks]**