QUESTION ONE (COMPULSORY) [30 MARKS]

a. State two main levels of a programming language. low level and High level [2 marks]

b. What kinds of knowledge can be represented in propositional logic? [1 marks]

c. By defining the aspects of logic, differentiate between **logic** and **logic programming** [3 marks]

d. Explain why Logic Programming is important and powerful. [3 marks]

e. Briefly explain the difference between computation and deduction and explain the connection of the two to logic programming. [3 marks]

f. What is symbolic logic? Using relevant arguments give a general pattern used in representing symbolic logic. [3 marks]

g. Explain the meaning of the following. [2 marks]

$$\frac{A \quad B}{C}$$

h. Using relevant arguments, differentiate between Modus Ponens and Modus Tollens ~~using relevant arguments~~. [4 marks]

i. Explain how lists are handled in prolog [2 marks]

j. Write a prolog program or database of facts and rules that:

i. Concatenate two lists [2 marks]

ii. Find the total cost of list of items [5 marks]

QUESTION TWO [20 MARKS]

a. What is the role of a **don't** care symbol (_) in prolog programing. -pattern matching & variable naming [1 marks]

b. Using a prolog syntax or statement, explain:

i. Conjunction of goals in a query ; father(X,Y) :- parent(X,Y), male(X) [2 marks]

ii. Disjunction of a goal in a query ; child_of(X,Y) :- father(X,Y); mother(X,Y) [2 marks]

iii. Backtracking process. a procedure in which prolog reaches the truth value of different predicates by checking whether they are correct or not [2 marks]

c. State any two common fallacies in logic reasoning. [2 marks]

d. Explain the connection between computation and deduction reasoning. [3 marks]

e. Discuss the concept of conflict resolution and its implementation in predicate logic. [4 marks]

f. Simulate the output of the following goals. [4 marks]

Vague/ambiguous
-Arguing the question
-Jumping to conclusions

Facts - A fact is a predicate that is true
Rules - extinctions of facts that contain conditional clauses

Atoms - strings made up of lower and uppercase letters, digits and the underscore, starting with a lowercase letter
elephant, b, abcXYZ

?- X is Y+1, Y=3. X=4, Y=3

?- X=:=3+2. X=:=5 false since X=4 which is not equal to 5

?- X is between (101,120). false · X=4, which is not in the specified range

?- 4+1=:=3+2. true 5=5

5 5

[20 MARKS]

QUESTION THREE

a. What is the relationship between judgment and proof? **[2 marks]**

b. Explain, using appropriate illustrations the following system inference strategies.

i. Goal driven system/backward chaining **[2 marks]**

ii. Data driven system/forward chaining **[2 marks]**

c. In the context of propositional logic and predicate calculus, explain the meaning of :

i. Alphabets **[2 marks]**

ii. Well-formed-formulas (wffs) **[2 marks]**

iii. Atomic formula **[2 marks]**

d. Explain condition under which the cut operator or function (!) can be used. **[2 marks]**

e. Explain the meaning of the following operator by giving a prolog syntax.

i. =:= - arithmetic comparison operator used to check whether two arithmetic expressions are equal. Numeric equality operator 3+2=:=5 true **[2 marks]**

ii. =\= - arithmetic comparison operator used to check whether two arithmetic expressions are not equal 3+2=\=5 false **[2 marks]**

iii. ln. - natural logarithmic function which represent the power to which the mathematical constant 'e' must be raised to obtain the value X ln(X). **[2 marks]**

[20 MARKS]

QUESTION FOUR

a. Explain various characteristics of a prolog program - Logical inference - Declarative nature - pattern matching - backtracking **[3 marks]**

b. Prolog syntax is based on part of predicate calculus known as the Horn Clause. Explain the meaning of Horn clause using relevant illustrations. **[2 marks]**

Horn clause logic variable is unification Recursion

c. Explain how the following processes are handled in logic programming.

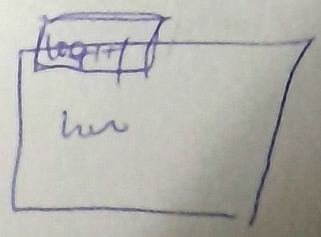
i. Resolution **[2 marks]**

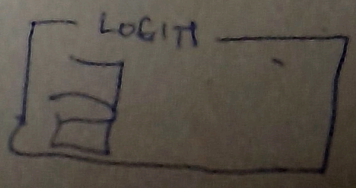
ii. Unification **[2 marks]**

iii. Instantiation **[2 marks]**

d. Consider the prolog program bellow.

```
sum :- readint(X), readint(Y), sum is X+Y, write(sum), ln.
```





i. Write an appropriate goal or query for the program and simulate its output(s) **[2 marks]**

e. By differentiating between tail recursion and non-tail recursion, explain how recursion is handled in prolog programs. **[3 marks]**

f. Write a program to find the power of any number using tail recursion. **[4 marks]**

QUESTION FIVE **[20 MARKS]**

a. Using appropriate syntax, explain the following elements of a prolog program.

i. Fact - a statement that represents a piece of information about a relationship btwn items. facts are used to describe the initial state of the world or to define basic relationships parent(John, mary). **[2 marks]**

ii. Rule - A statement that defines a logical relationship or condition based on a given premises (facts). mammal(X):-animal(X), has_hair(X) **[2 marks]**

iii. Query - A statement that poses a question or asks for info: parent(John, mary). **[2 marks]**

b. Consider the prolog program below that finds the factorial of a positive integer number (N).

```
% Domain: I=Integer
% Predicate: fact (N, F).
% clauses:
fact(0,1).
fact(1,1).
fact(N, F):- N1 is N-1, fact(N1, F1), F is N*F1.
```

i. Explain how the above program will be consulted. **[2 marks]**

ii. Write a query or a goal that will output the factorial of a number 6. **[2 marks]**

iii. Explain how the prolog compiler will arrive on the output stated in **(ii)** above. **[2 marks]**

iv. Can this program allows backtracking process? Explain. **[2 marks]**

c. Explain the meaning of **ontology engineering** and **ontology language** **[2 marks]**

d. Discuss briefly any **FOUR** types of reasoning systems as used in logic programming. **[4 marks]**

Ontology engineering refers to the process of creating, designing and maintaining ontologies
ontology refers a structured and organized rep. of knowledge about a particular domain capturing concepts, rel., properties and constraints that exists within that domain.

Ontology language is a formal language designed for creating, rep. and querying ontologies

i) Deductive reasoning
conclusions are derived from established premises using logical rules. this can be applied through backward chaining.

ii) Inductive reasoning
involves generalizing from specific instances to form broader hypotheses or conclusions. can be implemented through techniques like induction over lists or recursive definitions

iii) Abductive Reasoning
involves generating explanations or hypotheses to explain observed phenomena or facts. It is used when there is incomplete or uncertain information

iv) Analogical reasoning
Involves drawing conclusions based on similarities btwn situations or cases. It can be used to transfer knowledge from one domain to another by identifying and exploiting similarities