What is Prolog?

Prolog is a logic programming language used for solving problems that involve reasoning and knowledge representation.

Why is Prolog Used?

Artificial Intelligence (AI) Applications

- Used in Natural Language Processing (NLP).
- Develops Expert Systems that simulate human decision-making

Database Querying

- Efficient for searching and managing data.

Modeling and Planning

Helps create models for complex scenarios.

Educational Tools

- Simplifies teaching logic and reasoning concepts.

Applications of Prolog

- **Puzzles and Games**
  - Used to solve logic puzzles and create games.

Medical Diagnosis Systems

Helps identify diseases based on symptoms.

Robotics and Automation

- Assists in automated decision-making for robots.

Natural Language Processing (NLP)

- Understands and processes human languages.

In Prolog, a **fact** is a basic statement that declares a relationship between entities. Here's how you write the given relationship as a fact:

Syntax for Writing the Fact:

likes(john, mary).

Explanation:

1. **likes**: This is the predicate (relationship or action).
2. **john**: The first argument (subject of the relationship).
3. **mary**: The second argument (object of the relationship).
4. **.**: The period marks the end of the fact.
5. Here are some additional facts you can use to expand the knowledge base:

6. **Example Facts:**

```
7.prolog
8.CopyEdit
9.likes(john, mary).       % John likes Mary
10.likes(mary, ice_cream).  % Mary likes ice cream
11.likes(susan, books).     % Susan likes books
12.likes(alex, music).      % Alex likes music
13.likes(paul, football).   % Paul likes football
14.likes(anna, painting).   % Anna likes painting
15.likes(john, pizza).      % John likes pizza
16.likes(susan, hiking).    % Susan likes hiking
17.likes(mary, dancing).    % Mary likes dancing
18.likes(alex, coding).     % Alex likes coding
```

Explanation:

- Each **fact** consists of a predicate (likes) and two arguments.
- The arguments can represent people, objects, or concepts.
- You can define any relationships to build a more detailed knowledge base.

How to Use These Facts:

1. **Querying Relationships:**
    - To check if John likes Mary:

```
?- likes(john, mary).
```

This will return true.

Facts for Relationships and Attributes

1. Likes Relationship

prolog

CopyEdit

```
likes(john, mary).
likes(susan, chocolate).
likes(alex, movies).
likes(paul, football).
likes(mary, traveling).
```

2. Dislikes Relationship

prolog

CopyEdit

```
dislikes(john, spiders).
dislikes(susan, noise).
dislikes(paul, rainy_days).
dislikes(mary, homework).
```

3. Friendship

prolog

CopyEdit

```
friends(john, alex).
friends(susan, mary).
friends(alex, paul).

friends(john, susan).
```

4. Family Relationships

prolog

CopyEdit

```
parent(john, susan).

parent(mary, paul).

parent(alex, anna).

sibling(susan, paul).

sibling(john, alex).
```

5. Occupation

prolog

CopyEdit

```
occupation(john, doctor).

occupation(mary, teacher).

occupation(alex, engineer).

occupation(paul, artist).

occupation(susan, scientist).
```

6. Owns/Has

prolog

CopyEdit

```
owns(john, car).

owns(mary, house).

owns(alex, laptop).

owns(paul, bike).

owns(susan, garden).
```

7. Animal Relationships

prolog

CopyEdit

```
has_pet(john, dog).
has_pet(mary, cat).
has_pet(alex, rabbit).
has_pet(susan, parrot).
```

8. Hobbies

prolog

CopyEdit

```
hobby(john, reading).
hobby(mary, painting).
hobby(susan, cycling).
hobby(alex, programming).
hobby(paul, photography).
```