Objective: Deepen participants' understanding of Prolog programming and explore advanced concepts and techniques.

Duration: 7-8 hours (including breaks)

1. One-Day Prolog Programming Workshop Plan:

Session1. Prolog Syntax and Structure (60 minutes)

- Explain the basics of Prolog, and its applications.
- Introduce participants to Prolog syntax, including facts and rules.
- Explain the use of variables, atoms, and predicates.
- Demonstrate the use of Prolog's logical operators (e.g., conjunction, disjunction, negation).
- Walk through examples to illustrate the structure of Prolog programs.

Session2. Prolog Knowledge Base (30 minutes)

- Explain the concept of a knowledge base in Prolog.
- Demonstrate how to define facts and rules in the knowledge base.
- Guide participants through exercises to create their own knowledge base.

Session3. Recursive Rules and Backtracking (60 minutes)

- Discuss the concept of recursion in Prolog.
- Demonstrate how to define recursive rules.
- Explain the mechanism of backtracking in Prolog.
- Engage participants in exercises to implement recursive rules and understand backtracking.

Session4. Advanced Prolog Concepts (60 minutes)

- Briefly introduce some advanced topics in Prolog, such as cuts, lists, and built-in predicates.
- Discuss handling input/output operations in Prolog.
- Explain how to read from and write to files in Prolog.

Lunch Break (30 mins)

Session5. Prolog Lists and Arithmetic (60 minutes)

- Explain Prolog's list data structure and its built-in predicates.
- Demonstrate how to manipulate lists, including appending, selecting, and sorting.
- Introduce arithmetic operations in Prolog and how to use them effectively.

Session6. Hands-on Programming Exercises (90 minutes)

- Implementing factorial and Fibonacci of a given number in Prolog
- Solving the 4-Queen and 8-puzzle problem using Prolog
- Solving the traveling salesman problem using Prolog

Session7. Q&A (30 minutes)

- Encourage participants to ask questions and clarify any doubts.
- Closing remarks and share additional learning resources and references for further study.

Equipment Requirements:

- Computers or laptops for each participant.
- Internet access for downloading software and resources.
- Text editor or Prolog development environment.
- Web browser for accessing online resources and documentation.

Software Requirements:

- Prolog interpreter or compiler. Recommended options include:
    - SWI-Prolog (https://www.swi-prolog.org/)
    - GNU Prolog (http://www.gprolog.org/)

Note: Participants should have the necessary software installed and configured on their computers prior to the workshop.

2. Two-Day Prolog Programming Workshop Plan:

Day-1:

Session1. Prolog Syntax and Structure (60 minutes)

- Explain the basics of Prolog, and its applications.
- Introduce participants to Prolog syntax, including facts and rules.
- Explain the use of variables, atoms, and predicates.
- Demonstrate the use of Prolog's logical operators (e.g., conjunction, disjunction, negation).
- Walk through examples to illustrate the structure of Prolog programs.

Session2. Prolog Knowledge Base (30 minutes)

- Explain the concept of a knowledge base in Prolog.
- Demonstrate how to define facts and rules in the knowledge base.
- Guide participants through exercises to create their own knowledge base.

Session3. Querying the Knowledge Base (60 minutes)

- Explain the process of querying the Prolog knowledge base.
- Introduce the concept of unification and pattern matching.
- Demonstrate how to query the knowledge base using built-in predicates.
- Conduct exercises for participants to practice writing queries and interpreting results.

Session4. Recursive Rules and Backtracking (60 minutes)

- Discuss the concept of recursion in Prolog.
- Demonstrate how to define recursive rules.
- Explain the mechanism of backtracking in Prolog.
- Engage participants in exercises to implement recursive rules and understand backtracking.

Lunch Break (30 mins)

Session5. Advanced Prolog Concepts (60 minutes)

- Briefly introduce some advanced topics in Prolog, such as cuts, lists, and built-in predicates.
- Discuss handling input/output operations in Prolog.
- Explain how to read from and write to files in Prolog.

Session6. Hands-on Programming Exercises (90 minutes)

- Writing simple facts using Prolog.
- Writing predicates to convert centigrade temperatures to Fahrenheit and check if a temperature is below freezing.
- Implementing factorial and Fibonacci of a given number in Prolog.
- Solving the Monkey Banana problem using Prolog.

Session7. Wrap-up and Q&A (30 minutes)

- Encourage participants to ask questions and clarify any doubts.
- Share additional learning resources and references for further study.

Day-2:

Session1. Prolog Lists and Arithmetic (60 minutes)

- Explain Prolog's list data structure and its built-in predicates.
- Demonstrate how to manipulate lists, including appending, selecting, and sorting.
- Introduce arithmetic operations in Prolog and how to use them effectively.

Session2. Hands-on Programming Exercises (60 minutes)

- Writing a program in Prolog for medical diagnosis and exploring the advantages and disadvantages of green and red cuts.
- Solving the 4-Queen and 8-puzzle problem using Prolog.

Session3. Problem Solving with Prolog (90 minutes)

- Solving the water jug problem using Prolog.
- Solving the traveling salesman problem using Prolog.

Lunch Break (30 mins)

Session4. Prolog AI Applications (60 minutes)

- Overview of Prolog's role in Artificial Intelligence.
- Prolog applications in Natural Language Processing (NLP).

Session5. Advanced Prolog AI Algorithms (90 minutes)

- Implementation of A* algorithm using Prolog.
- Implementation of Hill climbing algorithm using Prolog.

Session6. Q&A and Wrap-up (30 minutes)

- Encourage participants to ask questions and clarify any doubts.
- Closing remarks and share additional learning resources and references for further study.

Equipment Requirements:

- Computers or laptops for each participant.
- Internet access for downloading software and resources.
- Text editor or Prolog development environment.
- Web browser for accessing online resources and documentation.

Software Requirements:

- Prolog interpreter or compiler. Recommended options include:
    - SWI-Prolog (https://www.swi-prolog.org/)
    - GNU Prolog (http://www.gprolog.org/)

Note: Participants should have the necessary software installed and configured on their computers prior to the workshop.