

Some of the basic elements of Prolog are:

1. Term
2. Fact statements
3. Rule statements

1. Term: A Prolog term is a constant, a variable, or a structure.

A constant is either an atom or an integer.

Atoms are the symbolic values of Prolog and are similar to their counterparts in LISP. In particular, an atom is either a string of letters, digits, and underscores that begins with a lowercase letter or a string of any printable ASCII characters delimited by apostrophes.

2. Fact statements: A fact statement is simply a proposition that is assumed to be true.

For example:

- `female(khushi).` , it states Khushi is a female.
- `mother(varsha, khushi).` , it states Varsha is Mother of Khushi.

3. Rule Statements: Rule statements state rules of implication between propositions. For example: `ancestor(varsha, khushi) :- mother(varsha, khushi).`, it states that if Varsha is the mother of Khushi, then Varsha is an ancestor of Khushi.

Related posts:

1. Sequence Control & Expression | PPL
2. PPL:Named Constants
3. Parse Tree | PPL | Prof. Jayesh Umre
4. Loops | PPL | Prof. Jayesh Umre
5. Subprograms Parameter passing methods | PPL | Prof. Jayesh Umre

6. Programming Paradigms | PPL | Prof. Jayesh Umre
7. Subprograms Introduction | PPL | Prof. Jayesh Umre
8. Phases of Compiler | PPL | Prof. Jayesh Umre
9. Parse Tree | PPL
10. Influences on Language design | PPL | Prof. Jayesh Umre
11. Fundamentals of Subprograms | PPL | Prof. Jayesh Umre
12. Programming Paradigm
13. Influences on Language Design
14. Language Evaluation Criteria
15. OOP in C++ | PPL
16. OOP in C# | PPL
17. OOP in Java | PPL
18. PPL: Abstraction & Encapsulation
19. PPL: Semaphores
20. PPL: Introduction to 4GL
21. PPL: Variable Initialization
22. PPL: Conditional Statements
23. PPL: Array
24. PPL: Strong Typing
25. PPL: Coroutines
26. PPL: Exception Handler in C++
27. PPL: OOP in PHP
28. PPL: Character Data Type
29. PPL: Exceptions
30. PPL: Heap based storage management
31. PPL: Primitive Data Type
32. PPL: Data types

33. Programming Environments | PPL
34. Virtual Machine | PPL
35. PPL: Local referencing environments
36. Generic Subprograms
37. Local referencing environments | PPL | Prof. Jayesh Umre
38. Generic Subprograms | PPL | Prof. Jayesh Umre
39. PPL: Java Threads
40. PPL: Loops
41. PPL: Exception Handling
42. PPL: C# Threads
43. Pointer & Reference Type | PPL
44. Scope and lifetime of variable
45. Design issues for functions
46. Parameter passing methods
47. Fundamentals of sub-programs
48. Subprograms
49. Design issues of subprogram
50. Garbage Collection
51. Issues in Language Translation
52. PPL Previous years solved papers
53. Type Checking | PPL | Prof. Jayesh Umre
54. PPL RGPV May 2018 solved paper discussion| Prof. Jayesh Umre
55. PPL Viva Voce
56. PPL RGPV June 2017 Solved paper | Prof. Jayesh Umre
57. Concurrency
58. Basic elements of Prolog
59. Introduction and overview of Logic programming

60. Application of Logic programming
61. PPL: Influences on Language Design
62. Language Evaluation Criteria PPL
63. PPL: Sequence Control & Expression
64. PPL: Programming Environments
65. PPL: Virtual Machine
66. PPL: Programming Paradigm
67. PPL: Pointer & Reference Type
68. try-catch block in C++