



SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107

Accredited by NAAC-UGC with 'A' Grade

Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Artificial Intelligence and

Data Science

Course Name – Introduction to Artificial Intelligence

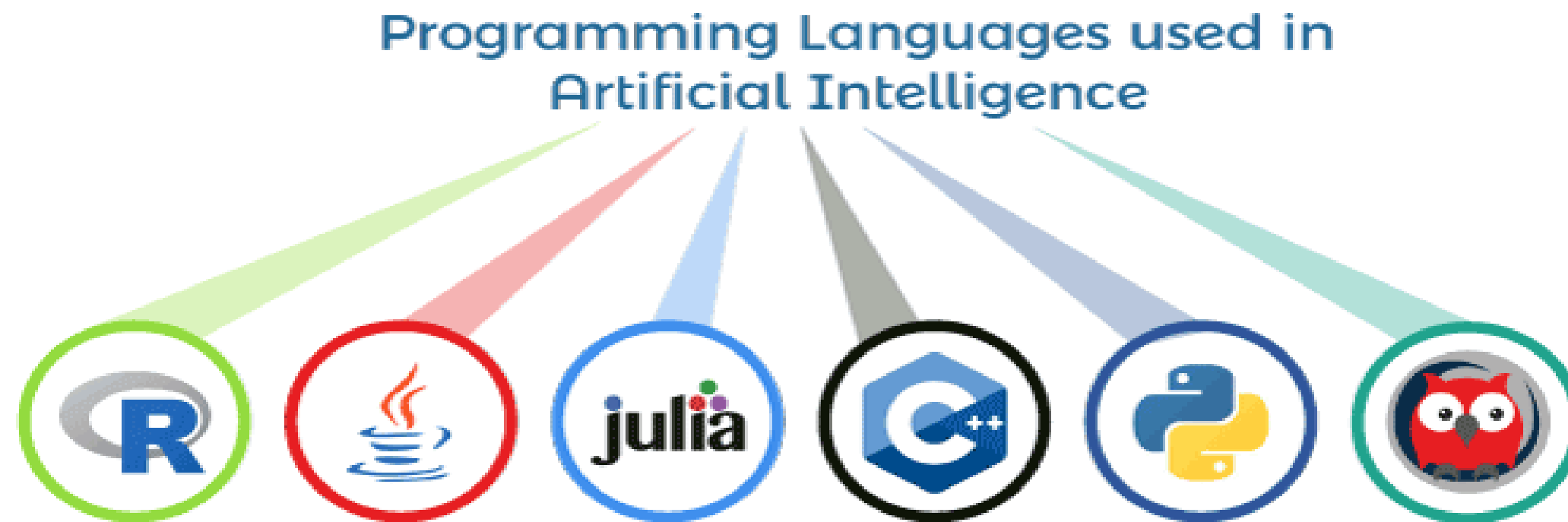
II Year / III Semester

Unit 1 LISP and PROLOG



Languages used in Artificial Intelligence

Artificial Intelligence has become an important part of human life as we are now highly dependent on machines. **Artificial Intelligence is a very important technology to develop and build new computer programs and systems, which can be used to simulate various intelligence processes like learning, reasoning, etc.**





Lisp

Lisp, an acronym for *list processing*, is a functional programming language that was designed for easy manipulation of data strings. As one of the oldest programming languages still in use, Lisp offers several different dialects and has influenced the development of other languages.

A unique feature of early Lisp versions compared to most other programming languages is that the code could be directly interpreted without a compiler. The source code itself could be parsed and interpreted directly on a system. Today, however, most Lisp versions require that code be compiled and then loaded into an image to run. This offers faster program execution speeds compared to direct interpretation.

What are the different dialects of Lisp?

Since its inception, Lisp has gone through multiple changes and iterations depending on the need. These language implementations are known as dialects -- many of which are open source. The most used Lisp-based languages today include the following:

- Clojure
- Emacs Lisp
- Common Lisp
- Julia
- Racket
- Scheme



In Lisp, all computation is expressed as a function of at least one object. Objects can be other functions, data items -- such as constants or variables -- or data structures. Lisp's ability to compute with symbolic expressions rather than numbers makes it convenient for artificial intelligence (AI) applications. While it isn't as popular as C, Python or Perl, Lisp is still used for AI programming as well as several other functions. Lisp continues to be popular in higher education, as students learn Lisp programming tactics and extend this knowledge to the private sector after graduation.

Benefits of Lisp

While there are several reasons why the Lisp programming language is still popular after all these years, perhaps the most important is that it's considered to be a relatively simple language to learn. This is probably why it's still popular in academia. Other benefits include the following:

- access to powerful and easy-to-integrate macros;
- the language itself is programmable to meet nearly any need;
- operates on most platforms; and
- many find programming in Lisp to be faster with smaller code footprints.



Who uses Lisp?

Lisp is used within academia for a variety of functions, ranging from basic programming and AI to machine learning and quantum computing. Outside the university walls, Lisp dialects are used by the following:

- symbolic AI programmers;
- quantum computing professionals;
- embedded systems programmers;
- those seeking a quick scripting language; and
- small or understaffed programming teams.



What is Prolog?

Prolog is a programming language that is well-suited for developing logic-based artificial intelligence applications. It is a declarative programming language, meaning that it allows the programmer to specify the rules and facts about a problem domain, and then the Prolog interpreter will use these rules and facts to automatically infer solutions to problems.

What Makes Prolog Different?

One of the key features of Prolog is its ability to handle uncertain or incomplete information. In Prolog, a programmer can specify a set of rules and facts that are known to be true, but they can also specify rules and facts that might be true or false. The Prolog interpreter will then use these rules and facts to automatically reason about the problem domain and find solutions that are most likely to be correct, given the available information. **Basics Prolog Program**

In Prolog, programs are made up of two main components: facts and rules. Facts are statements that are assumed to be true, such as “John is a man” or “the capital of France is Paris.” Rules are logical statements that describe the relationships between different facts, such as “if John is a man and Mary is a woman, then John is not Mary.”

Prolog programs are written using a syntax that is similar to natural language. For example, a simple Prolog program might look like this:

```
man(john).  
woman(mary).  
capital_of(france, paris).
```

```
not(X,Y) :- man(X), woman(Y).
```

In this example, the first three lines are facts, while the fourth line is a rule. The rule uses the not/2 predicate to state that if X is a man and Y is a woman, then X is not Y.



Things to Remember when using Prolog

There is no single “syntax” for Prolog, as the language allows for a wide range of different programming styles and approaches. However, here are some basic elements of Prolog syntax that are commonly used:

- **Facts** are statements that are assumed to be true. In Prolog, facts are written using a predicate name followed by a list of arguments enclosed in parentheses. For example: `man(john)`.
- **Rules** are logical statements that describe the relationships between different facts. In Prolog, rules are written using the predicate name followed by a list of arguments enclosed in parentheses, followed by a colon and a hyphen (`:-`) and the body of the rule. For example: `not(X,Y) :- man(X), woman(Y)`.
- **Variables** are used to represent values that can change or be determined by the interpreter. In Prolog, variables are written using a name that begins with an uppercase letter. For example: `X`
- **Queries** are used to ask the interpreter to find solutions to problems based on the rules and facts in the program. In Prolog, queries are written using the same syntax as facts, followed by a question mark (`?`). For example: `not(john, mary)?`



Applications of Prolog

The applications of prolog are as follows:

- Specification Language
- Robot Planning
- Natural language understanding
- Machine Learning
- Problem Solving
- Intelligent Database retrieval
- Expert System
- Automated Reasoning

Uses of the Prolog programming

The prolog programming mostly usable in artificial intelligence technology.

This programming language uses for pattern matching using the parse tree feature.

The prolog is used in the computation schematics.

This programming uses in the problem-solving and understanding of the natural language.

The prolog programming language uses planning and designing of the robot.

The prolog is used in the automation system and theorem proving with a variety of constraints.

The prolog is a useful language in machine learning and the graphical user interface.

This is used for expert system and term rewriting using arithmetic and symbolic constraints.