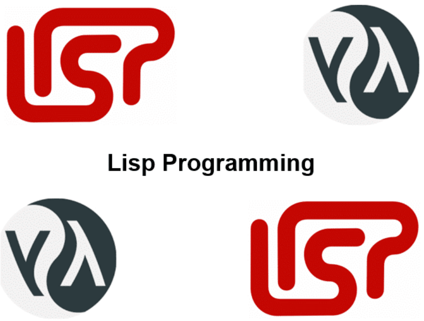
Lisp Programming

Duration: 3 days

Price: € 1999

Open Schedule



Audience Course Lisp Programming

The course Lisp Programming is intended for anyone who wants to learn programming in the functional programming language Lisp.

Prerequisites Course Lisp Programming

In order to participate in this course basic knowledge of programming in another programming language is beneficial to understanding but is not required.

Realization Training Lisp Programming

The theory is discussed on the basis of presentation slides. The theory is further explained through demos. After discussing a module there is the possibility to practice. Course times are from 9.30 to 16.30.

Certification Lisp Programming

After successful completion of the course, participants receive an official certificate Lisp Programming.

Content Course Lisp Programming

In the course Lisp Programming participants learn to develop applications in the functional programming language Lisp. Lisp is one of the oldest programming languages, but has been modernized to a significant degree in the course of its existence. Lisp is a machine independent language and is ideally suited for Artificial Intelligence software, because Lisp can process symbolic information effectively. The most commonly used Lisp dialects today are Common Lisp and Scheme .

Lisp Intro

The course starts with a discussion of the fundamentals of functional programming and expression evaluation. Next the program structure of Lisp applications, symbolic expressions and the macro system are discussed. Attention is also paid to atoms, lists and strings.

Lisp Syntax

Like any programming language, Lisp has variables, types and various control flow constructs. These are discussed in the module language syntax.

Functions

And also functions in Lisp are treated with the defun macro, with optional, rest and keyword parameters and with return values. Attention is also paid to typical functional functions such as lambdas and mapping functions.

Data Structures

Subsequently the various data structures that Lisp has to offer are on the course schedule such as sequences, lists and arrays, record structures, property lists, trees, hashtables and sets. The operations on these data structures such as unions and intersections between sets are also discussed.

Lisp I/O

And also input and output in Lisp applications is treated. This covers Stream I/O, input and output functions and reading and writing from and to files.

Classes and Structures

Finally the course concludes with a discussion of classes and structures in Lisp. The bundling of code in packages and the reuse of code by means of inheritance are also treated.

Modules Course Lisp Programming

Module 1 : Lisp Intro	Module 2 : Language Syntax	Module 3 : Functions
Functional Programming	List Forms	Defining Functions
Lisp Roots	Naming Conventions	defun Macro
Lisp Dialects	Scalar types	Parameter Passing
Common Lisp and Scheme	Numbers and Characters	Optional Parameters
Machine Independence	Symbols	Rest Parameters
Expression Evaluation	typep Predicate	&rest Symbol
Macro System	type-of Function	Keyword Parameters
Lisp Executer	Variables	Return Values
CLISP Compiler	setq and defvar	return-from Operator
Program Structure	Operators and Control Flow	Lambda Functions
Symbolic Expressions	cond, case, when	lambda Expression
Atoms, lists and strings	loop for, dotimes, dolist	Mapping Functions
Module 4 : Data Structures	**Module 5 : Lisp I/O**	**Module 6 : Structures and Classes**
Arrays and Indexing	I/O Streams	Defining Structures
Strings	Reading Input	defstruct Macro
Sequences	Input Functions	Access Functions
Sequence Functions	Output Functions	Constructors
Lists	Formatted Output	Predicates
cons Record Structure	File I/O	Copier Function
Symbols	Opening Files	Package as Namespace
Property Lists	:element-type Keyword	Creating Packages
Vectors	:external-format Argument	Using Packages
Fill Pointer	with-open-file	*package* Variable
Sets	:direction Keyword	Creating Classes
Unions and Intersections	:output Keyword	defmethod Macro
Trees and Hashtables	Reading and Writing	Inheritance