

---

---

**Information technology — Programming  
languages, their environments and  
system software interfaces —  
Programming language ISLISP**

*Technologies de l'information — Langages de programmation, leurs  
environnements et interfaces de logiciel système — Langage de  
programmation ISLISP*

**PDF disclaimer**

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.



**COPYRIGHT PROTECTED DOCUMENT**

© ISO/IEC 2007

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
Case postale 56 • CH-1211 Geneva 20  
Tel. + 41 22 749 01 11  
Fax + 41 22 749 09 47  
E-mail [copyright@iso.org](mailto:copyright@iso.org)  
Web [www.iso.org](http://www.iso.org)

Published in Switzerland

## Contents

<b>1</b>	<b>Scope</b>	<b>1</b>
<b>2</b>	<b>Normative references</b>	<b>1</b>
<b>3</b>	<b>Compliance of ISLISP processors and text</b>	<b>2</b>
<b>4</b>	<b>Terms and definitions</b>	<b>2</b>
<b>5</b>	<b>Notation and conventions</b>	<b>7</b>
<b>6</b>	<b>Lexemes</b>	<b>9</b>
6.1	Separators . . . . .	10
6.2	Comments . . . . .	10
<b>7</b>	<b>Textual representation</b>	<b>10</b>
<b>8</b>	<b>Reserved identifiers</b>	<b>11</b>
<b>9</b>	<b>Errors</b>	<b>11</b>
9.1	Classes of error specification . . . . .	12
9.2	Pervasive error types . . . . .	12
<b>10</b>	<b>Classes</b>	<b>13</b>
10.1	Metaclasses . . . . .	13
10.2	Predefined classes . . . . .	15
10.3	Standard classes . . . . .	16
10.3.1	Slots . . . . .	16
10.3.2	Creating instances of classes . . . . .	16
<b>11</b>	<b>Scope and extent</b>	<b>17</b>
11.1	The lexical principle . . . . .	17
11.2	Scope of identifiers . . . . .	17
11.3	Some specific scope rules . . . . .	18
11.4	Extent . . . . .	18
<b>12</b>	<b>Forms and evaluation</b>	<b>19</b>
12.1	Forms . . . . .	19
12.2	Function application forms . . . . .	20
12.3	Special forms . . . . .	20
12.4	Defining forms . . . . .	21
12.5	Macro forms . . . . .	21
12.6	The evaluation model . . . . .	21
12.7	Functions . . . . .	22
12.8	Defining operators . . . . .	26
<b>13</b>	<b>Predicates</b>	<b>28</b>
13.1	Boolean values . . . . .	28
13.2	Class predicates . . . . .	28
13.3	Equality . . . . .	29
13.4	Logical connectives . . . . .	31
<b>14</b>	<b>Control structure</b>	<b>33</b>
14.1	Constants . . . . .	33
14.2	Variables . . . . .	34

# ISO/IEC 13816:2007(E)

14.3	Dynamic variables . . . . .	37
14.4	Conditional expressions . . . . .	38
14.5	Sequencing forms . . . . .	40
14.6	Iteration . . . . .	41
14.7	Non-local exits . . . . .	42
14.7.1	Establishing and invoking non-local exits . . . . .	42
14.7.2	Assuring data consistency during non-local exits . . . . .	46
<b>15</b>	<b>Objects</b>	<b>47</b>
15.1	Defining classes . . . . .	47
15.1.1	Determining the class precedence list . . . . .	50
15.1.2	Accessing slots . . . . .	51
15.1.3	Inheritance of slots and slot options . . . . .	51
15.2	Generic functions . . . . .	52
15.2.1	Defining generic functions . . . . .	52
15.2.2	Defining methods for generic functions . . . . .	54
15.2.2.1	Agreement on parameter specializers and qualifiers . . . . .	55
15.2.2.2	Congruent lambda-lists for all methods of a generic function . . . . .	55
15.2.3	Inheritance of methods . . . . .	55
15.3	Calling generic functions . . . . .	56
15.3.1	Selecting the applicable methods . . . . .	56
15.3.2	Sorting the applicable methods . . . . .	56
15.3.3	Applying methods . . . . .	57
15.3.3.1	Simple method combination . . . . .	57
15.3.3.2	Standard method combination . . . . .	57
15.3.4	Calling more general methods . . . . .	59
15.4	Object creation and initialization . . . . .	60
15.4.1	Initialize-object . . . . .	60
15.5	Class enquiry . . . . .	61
<b>16</b>	<b>Macros</b>	<b>61</b>
<b>17</b>	<b>Declarations and coercions</b>	<b>63</b>
<b>18</b>	<b>Symbol class</b>	<b>65</b>
18.1	Symbol names . . . . .	65
18.1.1	Notation for symbols . . . . .	65
18.1.2	Alphabetic case in symbol names . . . . .	66
18.1.3	nil and () . . . . .	67
18.2	Symbol properties . . . . .	67
18.3	Unnamed symbols . . . . .	68
<b>19</b>	<b>Number class</b>	<b>68</b>
19.1	Number class . . . . .	69
19.2	Float class . . . . .	77
19.3	Integer class . . . . .	80
<b>20</b>	<b>Character class</b>	<b>83</b>
<b>21</b>	<b>List class</b>	<b>84</b>
21.1	Cons . . . . .	84
21.2	Null class . . . . .	87
21.3	List operations . . . . .	87
<b>22</b>	<b>Arrays</b>	<b>91</b>

22.1 Array classes . . . . .	91
22.2 General arrays . . . . .	92
22.3 Array operations . . . . .	93
<b>23 Vectors</b>	<b>95</b>
<b>24 String class</b>	<b>96</b>
<b>25 Sequence functions</b>	<b>99</b>
<b>26 Stream class</b>	<b>102</b>
26.1 Streams to files . . . . .	103
26.2 Other streams . . . . .	105
<b>27 Input and output</b>	<b>107</b>
27.1 Argument conventions for input functions . . . . .	107
27.2 Character I/O . . . . .	107
27.3 Binary I/O . . . . .	111
<b>28 Files</b>	<b>112</b>
<b>29 Condition system</b>	<b>114</b>
29.1 Conditions . . . . .	115
29.2 Signaling and handling conditions . . . . .	115
29.2.1 Operations relating to condition signaling . . . . .	116
29.2.2 Operations relating to condition handling . . . . .	117
29.3 Data associated with condition classes . . . . .	118
29.3.1 Arithmetic errors . . . . .	118
29.3.2 Domain errors . . . . .	118
29.3.3 Parse errors . . . . .	118
29.3.4 Simple errors . . . . .	119
29.3.5 Stream errors . . . . .	119
29.3.6 Undefined entity errors . . . . .	119
29.4 Error identification . . . . .	120
<b>30 Miscellaneous</b>	<b>122</b>
<b>Index</b>	<b>123</b>

## Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75% of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 13816 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

This second edition cancels and replaces the first edition (ISO/IEC 13816:1997), which has been technically revised.

## Introduction

The programming language ISLISP is a member of the LISP family.

The following factors influenced the establishment of design goals for ISLISP:

1. A desire of the international LISP community to standardize on those features of LISP upon which there is widespread agreement.
2. The existence of the incompatible dialects COMMON-LISP, EULISP, LE-LISP, and SCHEME (mentioned in alphabetical order).
3. A desire to affirm LISP as an industrial language.

This led to the following design goals for ISLISP:

1. ISLISP shall be compatible with existing LISP dialects where feasible.
2. ISLISP shall have as a primary goal to provide basic functionality.
3. ISLISP shall be object-oriented.
4. ISLISP shall be designed with extensibility in mind.
5. ISLISP shall give priority to industrial needs over academic needs.
6. ISLISP shall promote efficient implementations and applications.

# Information technology — Programming languages, their environments and system software interfaces — Programming language ISLISP

## 1 Scope

This International Standard specifies syntax and semantics of the computer programming language ISLISP by specifying requirements for a conforming ISLISP processor and a conforming ISLISP text.

This International Standard does not specify:

- (a) the size or complexity of an ISLISP text that exceeds the capacity of any specific data processing system or the capacity of a particular processor, nor the actions to be taken when the corresponding limits are exceeded;
- (b) the minimal requirements of a data processing system that is capable of supporting an implementation of a processor for ISLISP;
- (c) the method of preparation of an ISLISP text for execution and the method of activation of this ISLISP text, prepared for execution;
- (d) the typographical presentation of an ISLISP text published for human reading;
- (e) extensions that might or might not be provided by the implementation.

## 2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

- ISO/IEC TR 10034:1990, *Guidelines for the preparation of conformity clauses in programming language standards*
- IEEE standard 754-1985. *Standard for binary floating point arithmetic*