

AS 3955—1991
ISO/IEC 9899: 1990

Australian Standard[®]

Programming languages—C

This Australian Standard was prepared by Committee IT/9, Information Systems—Vocabulary and Software. It was approved on behalf of the Council of Standards Australia on 9 May 1991 and published on 12 July 1991.

The following interests are represented on Committee IT/9:

Attorney-Generals Department
AUSTEL
Australian Bankers' Association
Australian Bureau of Statistics
Australian Chamber of Manufacturers
Australian Computer Society
Australian Computer Users Association
Department of Defence
Department of Finance
Landata
Telecom Australia
Universities and Colleges

Review of Australian Standards. To keep abreast of progress in industry, Australian Standards are subject to periodic review and are kept up to date by the issue of amendments or new editions as necessary. It is important therefore that Standards users ensure that they are in possession of the latest edition, and any amendments thereto.

Full details of all Australian Standards and related publications will be found in the Standards Australia Catalogue of Publications; this information is supplemented each month by the magazine 'The Australian Standard', which subscribing members receive, and which gives details of new publications, new editions and amendments, and of withdrawn Standards.

Suggestions for improvements to Australian Standards, addressed to the head office of Standards Australia, are welcomed. Notification of any inaccuracy or ambiguity found in an Australian Standard should be made without delay in order that the matter may be investigated and appropriate action taken.

AS 3955—1991

Australian Standard[®]

Programming languages—C

First published as AS 3955—1991.

PUBLISHED BY STANDARDS AUSTRALIA
(STANDARDS ASSOCIATION OF AUSTRALIA)
1 THE CRESCENT, HOMEBUSH, NSW 2140

ISBN 0 7262 6970 0

PREFACE

This Standard was prepared by the Standards Australia Committee on Information Systems—Vocabulary and Software. It is identical with and has been reproduced from ISO/IEC 9899:1990 *Programming languages—C*.

Under arrangements made between Standards Australia and the international Standards bodies, ISO and IEC, as well as certain other Standards organizations, users of this Australian Standard are advised of the following:

- (a) Copyright is vested in Standards Australia.
- (b) The number of this Standard is not reproduced on each page; its identity is shown only on the cover and title pages.

For the purposes of this Australian Standard, the ISO/IEC text should be modified as follows:

- (i) *Terminology* The words ‘Australian Standard’ should replace the words ‘International Standard’ wherever they appear.
- (ii) *References* The references to International Standards should be replaced by references to Australian Standards as follows:

<i>Reference to International Standard</i>	<i>Australian Standard</i>
ISO	AS
646 Information processing—ISO 7-bit coded character set for information interchange	1776 Information processing—7-bit coded character set for information interchange
4217 Codes for the representation of currencies and funds	—

© Copyright — STANDARDS AUSTRALIA

Users of Standards are reminded that copyright subsists in all Standards Australia publications and software. Except where the Copyright Act allows and except where provided for below no publications or software produced by Standards Australia may be reproduced, stored in a retrieval system in any form or transmitted by any means without prior permission in writing from Standards Australia. Permission may be conditional on an appropriate royalty payment. Requests for permission and information on commercial software royalties should be directed to the head office of Standards Australia.

Standards Australia will permit up to 10 percent of the technical content pages of a Standard to be copied for use exclusively in-house by purchasers of the Standard without payment of a royalty or advice to Standards Australia.

Standards Australia will also permit the inclusion of its copyright material in computer software programs for no royalty payment provided such programs are used exclusively in-house by the creators of the programs.

Care should be taken to ensure that material used is from the current edition of the Standard and that it is updated whenever the Standard is amended or revised. The number and date of the Standard should therefore be clearly identified.

The use of material in print form or in computer software programs to be used commercially, with or without payment, or in commercial contracts is subject to the payment of a royalty. This policy may be varied by Standards Australia at any time.

CONTENTS

	<i>Page</i>
1 Scope	7
2 Normative references	7
3 Definitions and conventions	8
4 Compliance	
5 Environment	11
5.1 Conceptual models	11
5.1.1 Translation environment	11
5.1.2 Execution environments	12
5.2 Environmental considerations	16
5.2.1 Character sets	16
5.2.2 Character display semantics	18
5.2.3 Signals and interrupts	18
5.2.4 Environmental limits	18
6 Language	24
6.1 Lexical elements	24
6.1.1 Keywords	25
6.1.2 Identifiers	25
6.1.3 Constants	31
6.1.4 String literals	36
6.1.5 Operators	37
6.1.6 Punctuators	38
6.1.7 Header names	38
6.1.8 Preprocessing numbers	39
6.1.9 Comments	39
6.2 Conversions	40
6.2.1 Arithmetic operands	40
6.2.2 Other operands	42
6.3 Expressions	44
6.3.1 Primary expressions	45
6.3.2 Postfix operators	45
6.3.3 Unary operator	49
6.3.4 Cast operator	51
6.3.5 Multiplicative operators	52
6.3.6 Additive operators	52
6.3.7 Bitwise shift operators	54
6.3.8 Relational operators	54
6.3.9 Equality operators	55
6.3.10 Bitwise AND operator	56
6.3.11 Bitwise exclusive OR operator	56
6.3.12 Bitwise inclusive OR operator	56
6.3.13 Logical AND operator	57
6.3.14 Logical OR operator	57
6.3.15 Conditional operator	57
6.3.16 Assignment operators	59
6.3.17 Comma operator	60
6.4 Constant expressions	61
6.5 Declarations	63
6.5.1 Storage-class specifiers	64
6.5.2 Type specifiers	64
6.5.3 Type qualifiers	70
6.5.4 Declarators	71

	<i>Page</i>
6.5.5	Type names 75
6.5.6	Type definitions 76
6.5.7	Initialization 77
6.6	Statements 81
6.6.1	Labeled statements 81
6.6.2	Compound statement, or block 81
6.6.3	Expression and null statements 82
6.6.4	Selection statements 83
6.6.5	Iteration statements 84
6.6.6	Jump statements 84
6.7	External definitions 87
6.7.1	Function definitions 87
6.7.2	External object definitions 87
6.8	Preprocessing directives 91
6.8.1	Conditional inclusion 92
6.8.2	Source file inclusion 93
6.8.3	Macro replacement 95
6.8.4	Line control 99
6.8.5	Error directive 99
6.8.6	Pragma directive 99
6.8.7	Null directive 100
6.8.8	Predefined macro names 100
6.9	Future language directions 101
6.9.1	External names 101
6.9.2	Character escape sequences 101
6.9.3	Storage-class specifiers 101
6.9.4	Function declarators 101
6.9.5	Function definitions 101
6.9.6	Array parameters 101
7	Library 102
7.1	Introduction 102
7.1.1	Definitions of terms 102
7.1.2	Standard headers 102
7.1.3	Reserved identifiers 103
7.1.4	Errors <code><errno.h></code> 103
7.1.5	Limits <code><float.h></code> and <code><limits.h></code> 104
7.1.6	Common definitions <code><stddef.h></code> 104
7.1.7	Use of library functions 105
7.2	Diagnostic <code><assert.h></code> 107
7.2.1	Program diagnostics 107
7.3	Character handling <code><ctype.h></code> 108
7.3.1	Character testing functions 108
7.3.2	Character case mapping functions 110
7.4	Localization <code><locale.h></code> 112
7.4.1	Locale control 113
7.4.2	Numeric formatting convention inquiry 114
7.5	Mathematics <code><math.h></code> 117
7.5.1	Treatment of error conditions 117
7.5.2	Trigonometric functions 117
7.5.3	Hyperbolic functions 119
7.5.4	Exponential and logarithmic functions 120
7.5.5	Power functions 121
7.5.6	Nearest integer, absolute value, and remainder functions 122
7.6	Nonlocal jumps <code><setjmp.h></code> 124
7.6.1	Save calling environment 124
7.6.2	Restore calling environment 125
7.7	Signal handling <code><signal.h></code> 126

	<i>Page</i>
7.7.1 Specify signal handling	126
7.7.2 Send signal	127
7.8 Variable arguments <stdarg.h>	128
7.8.1 Variable argument list access macros	128
7.9 Input/output <stdio.h>	130
7.9.1 Introduction	130
7.9.2 Streams	131
7.9.3 Files	132
7.9.4 Operations on files	133
7.9.5 File access functions	134
7.9.6 Formatted input/output functions	137
7.9.7 Character input/output functions	147
7.9.8 Direct input/output functions	150
7.9.9 File positioning functions	151
7.9.10 Error-handling functions	153
7.10 General utilities <stdlib.h>	155
7.10.1 String conversion functions	155
7.10.2 Pseudo-random sequence generation functions	159
7.10.3 Memory management functions	160
7.10.4 Communication with the environment	161
7.10.5 Searching and sorting utilities	163
7.10.6 Integer arithmetic functions	164
7.10.7 Multibyte character functions	165
7.10.8 Multibyte string functions	167
7.11 String handling <string.h>	168
7.11.1 String function conventions	168
7.11.2 Copying functions	168
7.11.3 Concatenation functions	169
7.11.4 Comparison functions	170
7.11.5 Search functions	171
7.11.6 Miscellaneous functions	174
7.12 Date and time <time.h>	176
7.12.1 Components of time	176
7.12.2 Time manipulation functions	176
7.12.3 Time conversion functions	178
7.13 Future library directions	182
7.13.1 Errors <errno.h>	182
7.13.2 Character handling <ctype.h>	182
7.13.3 Localization <locale.h>	182
7.13.4 Mathematics <math.h>	182
7.13.5 Signal handling <signal.h>	182
7.13.6 Input/output <stdio.h>	182
7.13.7 General utilities <stdlib.h>	182
7.13.8 String handling <string.h>	182
 Annexes	
A Bibliography	183
B Language syntax summary	184
B.1 Lexical grammar	184
B.2 Phrase structure grammar	188
B.3 Preprocessing directives	193
C Sequence points	195
D Library summary	196
D.1 Errors <errno.h>	196
D.2 Common definitions <stddef.h>	196
D.3 Diagnostics <assert.h>	196

	<i>Page</i>
D.4 Character handling <code><ctype.h></code>	196
D.5 Localization <code><locale.h></code>	196
D.6 Mathematics <code><math.h></code>	197
D.7 Nonlocal jumps <code><set jmp.h></code>	197
D.8 Signal handling <code><signal.h></code>	197
D.9 Variable arguments <code><stdarg.h></code>	198
D.10 Input/output <code><stdio.h></code>	198
D.11 General utilities <code><stdlib.h></code>	200
D.12 String handling <code><string.h></code>	201
D.13 Date and time <code><time.h></code>	201
E Implementation limits	202
F Common warnings	204
G Portability issues	205
G.1 Unspecified behavior	205
G.2 Undefined behavior	206
G.3 Implementation-defined behavior	210
G.4 Locale-specific behavior	213
G.5 Common extensions	214
Index	216

STANDARDS AUSTRALIA

Australian Standard
Programming languages—C

1 Scope

This International Standard specifies the form and establishes the interpretation of programs written in the C programming language.¹ It specifies

- the representation of C programs;
- the syntax and constraints of the C language;
- the semantic rules for interpreting C programs;
- the representation of input data to be processed by C programs;
- the representation of output data produced by C programs;
- the restrictions and limits imposed by a conforming implementation of C.

This International Standard does not specify

- the mechanism by which C programs are transformed for use by a data-processing system;
- the mechanism by which C programs are invoked for use by a data-processing system;
- the mechanism by which input data are transformed for use by a C program;
- the mechanism by which output data are transformed after being produced by a C program;
- the size or complexity of a program and its data that will exceed the capacity of any specific data-processing system or the capacity of a particular processor;
- all minimal requirements of a data-processing system that is capable of supporting a conforming implementation.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 646:1983, *Information processing — ISO 7-bit coded character set for information interchange*.

ISO 4217:1987, *Codes for the representation of currencies and funds*.

¹ This International Standard is designed to promote the portability of C programs among a variety of data-processing systems. It is intended for use by implementors and programmers. It is accompanied by a Rationale document that explains many of the decisions of the Technical Committee that produced it.