

Implementation of
ISO/IEC 11729:1994

Information technology — Programming languages — Generic package of primitive functions for Ada

ICS 35.060.00

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW



Document Sharing Hub

Committees responsible for this British Standard

The preparation of this British Standard was entrusted to Technical Committee IST/5, Programming languages, upon which the following bodies were represented:

- British Computer Society
- Department of Trade and Industry (National Engineering Laboratory)
- Electricity Association
- Hoskyns Group plc
- IBM United Kingdom Ltd.
- International Computers Limited
- Joint Information Systems Committee
- National Computer Users Forum
- National Computing Centre Ltd.
- Rutherford Appleton Laboratory
- SEMA Group Systems Ltd.

This British Standard, having been prepared under the direction of the Information Systems Technology Assembly (T/-), was published under the authority of the Standards Board and comes into effect on 15 May 1995

© BSI 01-2000

The following BSI references relate to the work on this standard:
Committee reference IST/5
Draft for comment 93/642376 DC

ISBN 0 580 24089 4

Amendments issued since publication

Amd. No.	Date	Comments

Contents

	Page
Committees responsible	Inside front cover
National foreword	ii
Foreword	iv
Text of ISO/IEC 11729	1

National foreword

This British Standard reproduces verbatim ISO/IEC 11729:1994 and implements it as the UK national standard.

This British Standard is published under the direction of the Information Systems Technology Assembly whose Technical Committee IST/5 has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible international committee any enquiries on interpretation, or proposals for change, and keep UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

NOTE International and European Standards, as well as overseas standards, are available from Customer Services, BSI, 389 Chiswick High Road, London W4 4AL.

A British Standard does not purport to include all the necessary provisions of a contract. Users of British Standards are responsible for their correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, pages i and ii, the ISO/IEC title page, pages ii to iv, pages 1 to 18 and a back cover.

This standard has been updated (see copyright date) and may have had amendments incorporated. This will be indicated in the amendment table on the inside front cover.

**Information technology — Programming
languages — Generic package of primitive
functions for Ada**

*Technologies de l'information — Langages de programmation, leurs
environnements et interfaces de logiciel de système — Ensemble
générique de fonctions primitives pour l'Ada*



Reference number
ISO/IEC 11729:1994(E)

Contents

	Page
Foreword	iv
Introduction	1
1 Scope	1
2 Normative reference	1
3 Subprograms provided	1
4 Instantiations	2
5 Implementations	2
6 Machine numbers and storable machine numbers	3
7 Denormalized numbers	4
8 Exceptions	4
9 Specifications of the subprograms	5
9.1 EXPONENT — Exponent of the Canonical Representation of a Floating-Point Machine Number	5
9.2 FRACTION — Signed Mantissa of the Canonical Representation of a Floating-Point Machine Number	5
9.3 DECOMPOSE — Extract the Components of the Canonical Representation of a Floating-Point Machine Number	5
9.4 COMPOSE — Construct a Floating-Point Machine Number from the Components of its Canonical Representation	6
9.5 SCALE — Increment/Decrement the Exponent of the Canonical Representation of a Floating-Point Machine Number	6
9.6 FLOOR — Greatest Integer Not Greater Than a Floating-Point Machine Number, as a Floating-Point Number	6
9.7 CEILING — Least Integer Not Less Than a Floating-Point Machine Number, as a Floating-Point Number	7
9.8 ROUND — Integer Nearest to a Floating-Point Machine Number, as a Floating-Point Number	7
9.9 TRUNCATE — Integer Part of a Floating-Point Machine Number, as a Floating-Point Number	7
9.10 REMAINDER — Exact Remainder Upon Dividing One Floating-Point Machine Number by Another	7
9.11 ADJACENT — Floating-Point Machine Number Next to One Floating-Point Machine Number in the Direction of a Second	7
9.12 SUCCESSOR — Floating-Point Machine Number Next Above a Given Floating-Point Machine Number	8
9.13 PREDECESSOR — Floating-Point Machine Number Next Below a Given Floating-Point Machine Number	8
9.14 COPY_SIGN — Transfer of Sign from One Floating-Point Machine Number to Another	8
9.15 LEADING_PART — Floating-Point Machine Number with its Mantissa (in the Canonical Representation) Truncated to a Given Number of Radix-Digits	9
Annex A (normative) Ada specification for GENERIC_PRIMITIVE_FUNCTIONS	10
Annex B (informative) Rationale	10
B.1 Introduction and motivation	10
B.2 History	11
B.3 Packaging	12
B.4 Implementation permissions	12
B.5 Accuracy requirements	12
B.6 Discussion of individual subprograms	14
B.7 Relationship to other standards	17
B.8 Influence on Ada 9X	17

Annex C (informative) Bibliography

Page
17

Descriptors: Data processing, programming (computers), computer software, programming languages, ADA, computer interfaces.

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. 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.

International Standard ISO/IEC 11729 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee 22, *Programming languages, their environments and system software interfaces*.

Annex A forms an integral part of this International Standard. Annex B and Annex C are for information only.

Introduction

The generic package described here is intended to provide primitive operations that are required to endow mathematical software, such as implementations of the elementary functions, with the qualities of accuracy, efficiency and portability. With this International Standard, such mathematical software can achieve all of these qualities simultaneously; without it, one or more of them typically must be sacrificed. The generic package specification included in this International Standard is presented as a compilable Ada specification in Annex A with explanatory text in numbered clauses in the main body of text. The explanatory text is normative, with the exception of notes.

The word “may” as used in this International Standard consistently means “is allowed to” (or “are allowed to”). It is used only to express permission, as in the commonly occurring phrase “an implementation may”; other words (such as “can,” “could” or “might”) are used to express ability, possibility, capacity or consequentiality.

In formulas, $\lfloor v \rfloor$ and $\lceil v \rceil$ mean the greatest integer less than or equal to v and the least integer greater than or equal to v , respectively, and other notations have their customary meaning.

1 Scope

This International Standard specifies primitive functions and procedures for manipulating the fraction part and the exponent part of machine numbers (see clause 6) of the generic floating-point type. Additional functions are provided for directed rounding to a nearby integer, for computing an exact remainder, for determining the immediate neighbors of a floating-point machine number, for transferring the sign from one floating-point machine number to another and for shortening a floating-point machine number to a specified number of leading radix-digits. Some subprograms are redundant in that they are combinations of other subprograms. This is intentional so that convenient calls and fast execution can be provided to the user.

These subprograms are intended to augment standard Ada operations and to be useful in portably implementing such packages as those providing real and complex elementary functions, where (for example) the steps of argument reduction and result construction demand fast, error-free scaling and remaindering operations.

This International Standard is applicable to programming environments conforming to ISO 8652:1987.

2 Normative reference

The following standard contains provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the edition indicated was 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 edition of the standard indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 8652:1987, *Programming languages — Ada*. (Endorsement of ANSI Standard 1815A-1983)

3 Subprograms provided

The following fifteen subprograms are provided:

EXPONENT	FRACTION	DECOMPOSE	COMPOSE	SCALE
FLOOR	CEILING	ROUND	TRUNCATE	REMAINDER
ADJACENT	SUCCESSOR	PREDECESSOR		
COPY_SIGN	LEADING_PART			