

INTERNATIONAL STANDARD

ISO/IEC 10967-1

First edition
1994-12-15

Information technology — Language independent arithmetic —

Part 1:

Integer and floating point arithmetic

*Technologies de l'information — Arithmétique indépendante de
langage —*

Partie 1: Arithmétique de nombres entiers et en virgule flottante



Reference number
ISO/IEC 10967-1:1994(E)

Contents

1	Scope	1
1.1	Specifications included in this part of ISO/IEC 10967	1
1.2	Specifications not within the scope of this part of ISO/IEC 10967	2
2	Conformity	3
3	Normative reference	3
4	Symbols and definitions	4
4.1	Symbols	4
4.2	Definitions	4
5	The arithmetic types	6
5.1	Integer types	7
5.1.1	Operations	8
5.1.2	Modulo integers versus overflow	8
5.1.3	Axioms	9
5.2	Floating point types	10
5.2.1	Range and granularity constants	11
5.2.2	Operations	11
5.2.3	Approximate operations	12
5.2.4	Approximate addition	12
5.2.5	Rounding	13
5.2.6	Result function	13
5.2.7	Axioms	14
5.2.8	Rounding constants	15
5.2.9	Conformity to IEC 559	16
5.3	Conversion operations	16
6	Notification	18
6.1	Notification alternatives	18
6.1.1	Language defined notification	18
6.1.2	Recording of indicators	19
6.1.3	Termination with message	20
6.2	Delays in notification	20
6.3	User selection of alternative for notification	21
7	Relationship with language standards	21
8	Documentation requirements	22

Annexes

A	Rationale	24
A.1	Scope	24
A.1.1	Specifications included in this part of ISO/IEC 10967	24
A.1.2	Specifications not within the scope of this part of ISO/IEC 10967	24
A.1.3	Proposed follow-ons to this part of ISO/IEC 10967	25
A.2	Conformity	25
A.2.1	Validation	26
A.3	Normative references	26
A.4	Symbols and definitions	26
A.4.1	Symbols	27
A.4.2	Definitions	27
A.5	The arithmetic types	28
A.5.1	Integer types	28
A.5.1.1	Operations	30
A.5.1.2	Modulo integers versus overflow	30
A.5.1.3	Axioms	30
A.5.2	Floating point types	31
A.5.2.1	Range and granularity constants	35
A.5.2.2	Operations	36
A.5.2.3	Approximate operations	37
A.5.2.4	Approximate addition	37
A.5.2.5	Rounding	39
A.5.2.6	Result function	39
A.5.2.7	Axioms	40
A.5.2.8	Rounding constants	40
A.5.2.9	Conformity to IEC 559	41
A.5.2.10	Relations among floating point types	41
A.5.2.11	Levels of predictability	41
A.5.2.12	Identities	42
A.5.2.13	Precision, accuracy, and error	44
A.5.2.14	Extra precision	47
A.5.3	Conversion operations	48
A.6	Notification	48
A.6.1	Notification alternatives	48
A.6.1.1	Language defined notification	49
A.6.1.2	Recording of indicators	49
A.6.1.3	Termination with message	50
A.6.2	Delays in notification	50
A.6.3	User selection of alternative for notification	51
A.7	Relationship with language standards	51
A.8	Documentation requirements	52
B	Partial conformity	54
C	IEC 559 bindings	55
C.1	Summary	55
C.2	Notification	56
C.3	Rounding	56

D	Requirements beyond IEC 559	57
E	Bindings for specific languages	58
E.1	General comments	58
E.2	Ada	59
E.3	BASIC	62
E.4	C	64
E.5	Common Lisp	67
E.6	Fortran	70
E.7	Modula-2	73
E.8	Pascal and Extended Pascal	73
E.9	PL/I	76
F	Example of a conformity statement	79
F.1	Types	79
F.2	Integer parameters	79
F.3	Floating point parameters	80
F.4	Definitions	80
F.5	Expressions	81
F.6	Notification	81
G	Example programs	82
G.1	Verifying platform acceptability	82
G.2	Selecting alternate code	82
G.3	Terminating a loop	83
G.4	Fast versus reliable	83
G.5	High-precision multiply	83
G.6	Estimating error	84
G.7	Saving and restoring indicators	84
H	Bibliography	86
J	Glossary	89

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 10967-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

ISO/IEC 10967 consists of the following parts, under the general title *Information technology — Language independent arithmetic*:

- Part 1: *Integer and floating point arithmetic*
- Part 2: *Mathematical procedures*
- Part 3: *Complex arithmetic and procedures*

Additional parts will specify other arithmetic types or operations.

Annexes A to J of this part of ISO/IEC 10967 are for information only.

Introduction

The aims

Programmers writing programs that perform a significant amount of numeric processing have often not been certain how a program will perform when run under a given language processor. Programming language standards have traditionally been somewhat weak in the area of numeric processing, seldom providing an adequate specification of the properties of arithmetic data types, particularly floating point numbers. Often they do not even require much in the way of documentation of the actual arithmetic data types by a conforming language processor.

It is the intent of this part of ISO/IEC 10967 to help to redress these shortcomings, by setting out precise definitions of integer and floating point data types, and requirements for documentation. This is done in a way that makes as few presumptions as possible about the underlying machine architecture.

It is not claimed that this part of ISO/IEC 10967 will ensure complete certainty of arithmetic behavior in all circumstances; the complexity of numeric software and the difficulties of analysing and proving algorithms are too great for that to be attempted. Rather, the requirements set forth here will provide a firmer basis than hitherto for attempting such analysis.

Hence the first aim of this part of ISO/IEC 10967 is to enhance the predictability and reliability of the behavior of programs performing numeric processing.

The second aim, which helps to support the first, is to help programming language standards to express the semantics of arithmetic data types. These semantics need to be precise enough for numerical analysis, but not so restrictive as to prevent efficient implementation of the language on a wide range of platforms.

The third aim is to help enhance the portability of programs that perform numeric processing across a range of different platforms. Improved predictability of behavior will aid programmers designing code intended to run on multiple platforms, and will help in predicting what will happen when such a program is moved from one conforming language processor to another.

Note that this part of ISO/IEC 10967 does not attempt to ensure bit-for-bit identical results when programs are transferred between language processors, or translated from one language into another. Programming languages and platforms are too diverse to make that a sensible goal. However, experience shows that diverse numeric environments can yield comparable results under most circumstances, and that with careful program design significant portability is actually achievable.

The content

This part of ISO/IEC 10967 defines the fundamental properties of integer and floating point numbers. These properties are presented in terms of a parameterized model. The parameters allow enough variation in the model so that most platforms are covered, but when a particular set of parameter values is selected, and all required documentation is supplied, the resulting information should be precise enough to permit careful numerical analysis.

The requirements of this part of ISO/IEC 10967 cover three areas. First, the programmer must be given runtime access to the parameters and functions that describe the arithmetic properties of the platform. Second, the executing program must be notified when proper results cannot be returned (e.g., when a computed result is out of range or undefined). Third, the numeric properties of conforming platforms must be publicly documented.

This part of ISO/IEC 10967 focuses on the classical integer and floating point data types. Later parts will consider common mathematical procedures (part 2), complex numbers (part 3), and possibly additional arithmetic types such as fixed point.

Relationship to hardware

ISO/IEC 10967 is not a hardware architecture standard. It makes no sense to talk about an “LIA machine.” Future platforms are expected either to duplicate existing architectures, or to satisfy high quality architecture standards such as IEC 559 (also known as IEEE 754). The floating point requirements of this part of ISO/IEC 10967 are compatible with (and enhance) IEC 559.

This part of ISO/IEC 10967 provides a bridge between the abstract view provided by a programming language standard and the precise details of the actual arithmetic implementation.

The benefits

Adoption and proper use of this part of ISO/IEC 10967 can lead to the following benefits.

Language standards will be able to define their arithmetic semantics more precisely without preventing the efficient implementation of their language on a wide range of machine architectures.

Programmers of numeric software will be able to assess the portability of their programs in advance. Programmers will be able to trade off program design requirements for portability in the resulting program.

Programs will be able to determine (at run time) the crucial numeric properties of the implementation. They will be able to reject unsuitable implementations, and (possibly) to correctly characterize the accuracy of their own results. Programs will be able to extract apparently implementation dependent data (such as the exponent of a floating point number) in an implementation independent way. Programs will be able to detect (and possibly correct for) exceptions in arithmetic processing.

End users will find it easier to determine whether a (properly documented) application program is likely to execute satisfactorily on their platform. This can be done by comparing the documented requirements of the program against the documented properties of the platform.

Finally, end users of numeric application packages will be able to rely on the correct execution of those packages. That is, for correctly programmed algorithms, the results are reliable if and only if there is no notification.

This document is a preview generated by EVS

Annex A is intended to be read in parallel with the standard.

**Information technology –
Language independent arithmetic –
Part 1:
Integer and floating point arithmetic**

1 Scope

This part of ISO/IEC 10967 defines the properties of integer and floating point data types on computer systems to ensure that the processing of arithmetic data can be undertaken in a reliable and predictable manner. Emphasis is placed on documenting the existing variation between systems, not on the elimination of such variation. The requirements of this part of ISO/IEC 10967 shall be in addition to those that may be specified in other standards, such as those for programming languages (See clause 7).

It is not the purpose of this part of ISO/IEC 10967 to ensure that an arbitrary numerical function can be so encoded as to produce acceptable results on all conforming systems. Rather, the goal is to ensure that the properties of arithmetic on a conforming system are made available to the programmer.

Therefore, it is not reasonable to demand that a substantive piece of software run on every implementation that can claim conformity to this part of ISO/IEC 10967.

An implementor may choose any combination of hardware and software support to meet the specifications of this part of ISO/IEC 10967. It is the arithmetic environment, as seen by the user, that does or does not conform to the specifications.

The term *implementation* (of this part of ISO/IEC 10967) denotes the total arithmetic environment, including hardware, language processors, exception handling facilities, subroutine libraries, other software, and all pertinent documentation.

1.1 Specifications included in this part of ISO/IEC 10967

This part of ISO/IEC 10967 defines integer and floating point data types. Definitions are included for bounded, unbounded, and modulo integer types, as well as both normalized and denormalized floating point types.

The specification for an arithmetic type includes

- a) The set of computable values.
- b) The set of computational operations provided, including
 - 1) primitive operations (addition, subtraction, etc.) with operands of the same type,