INTERNATIONAL STANDARD

ISO/IEC 10967-1

Second edition
2012-07-15

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 10967-1:2012(E)

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
Web www.iso.org

Published in Switzerland

Contents

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 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.*

This second edition cancels and replaces the first edition (ISO/IEC 10967-1:1994), which has been technically revised.

ISO/IEC 10967-1 consists of the following parts, under the general title *Information technology — Language independent arithmetic:*

— *Part 1: Integer and floating point arithmetic*

— *Part 2: Elementary numerical functions*

— *Part 3: Complex integer and floating point arithmetic and complex elementary numerical functions*

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 datatypes, particularly floating point numbers. Often they do not even require much in the way of documentation of the actual arithmetic datatypes 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 datatypes, and requirements for documentation.

It is not claimed that this part of ISO/IEC 10967 will ensure complete certainty of arithmetic behaviour in all circumstances; the complexity of numeric software and the difficulties of analysing and proving algorithms are too great for that to be attempted.

The first aim of this part of ISO/IEC 10967 is to enhance the predictability and reliability of the behaviour 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 datatypes.

The third aim is to help enhance the portability of programs that perform numeric processing across a range of different platforms. Improved predictability of behaviour 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. 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. In addition, the IEC 60559 (IEEE 754) standard goes a long way to ensure bit-for-bit identical results, and in this second edition of this part of ISO/IEC 10967 the requirements are tightened (compared to the first edition) to approach those of IEEE 754.

The content

This part of ISO/IEC 10967 defines the fundamental properties of integer and floating point datatypes. These properties are presented in terms of a parameterised model. The parameters allow enough variation in the model so that several integer and floating point datatypes are covered. In particular, the IEC 60559 (IEEE 754) floating point datatypes, both those of radix 2 and those of radix 10, are covered, as well as integer datatypes, both unlimited and limited, for the latter both signed or unsigned, 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 four areas. First, the programmer must be given runtime access to the specified operations on values of integer or floating point datatype. Second, the programmer must be given runtime access to the parameters (and parameter functions) that describe the arithmetic properties of an integer or floating point datatype. Third, the executing program must be notified when proper results cannot be returned (e.g., when a

computed result may be out of range or undefined). Fourth, the numeric properties of conforming platforms must be publicly documented.

This part of ISO/IEC 10967 focuses on the classical integer and floating point datatypes. Subsequent parts considers common elementary numerical functions (Part 2), complex numerical numbers and complex elementary numerical functions (Part 3).

The benefits

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

For programming language standards it will be possible to define their arithmetic semantics more precisely without preventing the efficient implementation of the 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.

In programs one 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 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.

Information technology — Language independent arithmetic —

Part 1: Integer and floating point arithmetic

1 Scope

This part of ISO/IEC 10967 specifies properties of many of the integer and floating point datatypes available in a variety of programming languages in common use for mathematical and numerical applications.

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 datatypes. Rather, the goal is to ensure that the properties of the arithmetic on a conforming datatype 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 datatypes and operations on values of those datatypes, of the computing environment as seen by the programmer/user, that does or does not conform to the specifications.

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

1.1 Inclusions

This part of ISO/IEC 10967 provides specifications for properties of integer and floating point datatypes as well as basic operations on values of these datatypes. Specifications are included for bounded and unbounded integer datatypes, as well as floating point datatypes. Boundaries for the occurrence of exceptions and the maximum error allowed are prescribed for each specified operation. Also the result produced by giving a special value operand, such as an infinity or a NaN (not-a-number), is prescribed for each specified floating point operation.

This part of ISO/IEC 10967 provides specifications for:

a) The set of required values of the arithmetic datatype.

b) A number of arithmetic operations, including:

   1) comparison operations on two operands of the same type,

   2) primitive operations (addition, subtraction, etc.) with operands of the same type,

   3) operations that access properties of individual values,

4) conversion operations of a value from one arithmetic datatype to another arithmetic datatype, where at least one of the datatypes is conforming to this part of ISO/IEC 10967, and

5) numerals for all values specified by this part of ISO/IEC 10967 for a conforming datatype.

This part of ISO/IEC 10967 also provides specifications for:

c) The results produced by an included floating point operation when one or more argument values are IEC 60559 special values.

d) Program-visible parameters that characterise the values and certain aspects of the operations of an arithmetic datatype.

e) Methods for reporting arithmetic exceptions.

1.2 Exclusions

This part of ISO/IEC 10967 provides no specifications for:

a) Arithmetic and comparison operations whose operands are of more than one datatype. This part of ISO/IEC 10967 neither requires nor excludes the presence of such "mixed operand" operations.

b) An interval datatype, or the operations on such data. This part of ISO/IEC 10967 neither requires nor excludes such data or operations.

c) A fixed point datatype, or the operations on such data. This part of ISO/IEC 10967 neither requires nor excludes such data or operations.

d) A rational datatype, or the operations on such data. This part of ISO/IEC 10967 neither requires nor excludes such data or operations.

e) The properties of arithmetic datatypes that are not related to the numerical process, such as the representation of values on physical media.

f) The properties of integer and floating point datatypes that properly belong in programming language standards or other specifications. Examples include:

1) the syntax of numerals and expressions in the programming language, including the precedence of operators in the programming language,

2) the syntax used for parsed (input) or generated (output) character string forms for numerals by any specific programming language or library,

3) the presence or absence of automatic datatype coercions, and the consequences of applying an operation to values of improper type, or to uninitialised data,

4) the rules for assignment, parameter passing, and returning value.

NOTE – See Clause 7 and Annex D for a discussion of language standards and language bindings.

The internal representation of values is beyond the scope of this standard. E.g., the value of the exponent bias, if any, is not specified, nor available as a parameter specified by this part

of ISO/IEC 10967. Internal representations need not be unique, nor is there a requirement for identifiable fields (for sign, exponent, and so on).

Furthermore, this part of ISO/IEC 10967 does not provide specifications for how the operations should be implemented or which algorithms are to be used for the various operations.

2 Conformity

It is expected that the provisions of this part of ISO/IEC 10967 will be incorporated by reference and further defined in other International Standards; specifically in programming language standards and in binding standards.

A binding standard specifies the correspondence between one or more of the arithmetic datatypes, parameters, and operations specified in this part of ISO/IEC 10967 and the concrete language syntax of some programming language. More generally, a binding standard specifies the correspondence between certain datatypes, parameters, and operations and the elements of some arbitrary computing entity. A language standard that explicitly provides such binding information can serve as a binding standard.

When a binding standard for a language exists, an implementation shall be said to conform to this part of ISO/IEC 10967 if and only if it conforms to the binding standard. In the case of conflict between a binding standard and this part of ISO/IEC 10967, the specifications of the binding standard takes precedence.

When a binding standard requires only a subset of the integer or floating point datatypes provided, an implementation remains free to conform to this part of ISO/IEC 10967 with respect to other datatypes independently of that binding standard.

When a binding standard requires only a subset of the operations specified in this part of ISO/IEC 10967, an implementation remains free to conform to this part of ISO/IEC 10967 with respect to other datatypes and operations, independently of that binding standard.

When no binding standard exists, an implementation conforms to this part of ISO/IEC 10967 if and only if it provides one or more datatypes and operations that together satisfy all the requirements of Clauses 5 through 8 that are relevant to those datatypes and operations. The implementation shall then document the binding.

Conformity to this part of ISO/IEC 10967 is always with respect to a specified set of datatypes and set of operations. Under certain circumstances, conformity to IEC 60559 is implied by conformity to this part of ISO/IEC 10967.

An implementation is free to provide arithmetic datatypes and arithmetic operations that do not conform to this part of ISO/IEC 10967 or that are beyond the scope of this part of ISO/IEC 10967. The implementation shall not claim conformity to this part of ISO/IEC 10967 for such datatypes or operations.

An implementation is permitted to have modes of operation that do not conform to this part of ISO/IEC 10967. A conforming implementation shall specify how to select the modes of operation that ensure conformity. However, a mode of operation that conforms to this part of ISO/IEC 10967 should be the default mode of operation.

NOTES

1 Language bindings are essential. Clause 8 requires an implementation to supply a binding if no binding standard exists. See Annex C.7 for recommendations on the proper content of a binding standard, Annex E for an example of a conformity statement, and Annex D for suggested language bindings.

2 A complete binding for this part of ISO/IEC 10967 may include (explicitly or by reference) a binding for IEC 60559 as well. See 5.2.1 and Annex B.

3 It is not possible to conform to this part of ISO/IEC 10967 without specifying to which datatypes and set of operations, and modes of operation, conformity is claimed.

4 This part of ISO/IEC 10967 requires that certain integer operations are made available for a conforming integer datatype, and that certain floating point operations are made available for a conforming floating point datatype.

5 All the operations specified in this part of ISO/IEC 10967 for a datatype must be provided for a conforming datatype, in a conforming mode of operation for that datatype.

3 Normative references

The following referenced documents are indispensable for the application of this part of ISO/IEC 10967. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 60559, *Standard for floating-point arithmetic.*

4 Symbols and definitions

4.1 Symbols

For the purposes of this document, the following symbols are used.

4.1.1 Operators and relations

All prefix and infix operators have their conventional exact mathematical meaning. In particular, this document uses:

$\Rightarrow$ and $\Leftrightarrow$ for logical implication and equivalence
$+$, $-$, $/$, $|x|$, $\lfloor x \rfloor$, $\lceil x \rceil$, and round(x) on real values
$\cdot$ for multiplication on real values
$<$, $\leqslant$, $\geqslant$, and $>$ between real values
$=$ and $\neq$ between real as well as special values
max on non-empty upwardly closed sets of real values
min on non-empty downwardly closed sets of real values
$\cup$, $\cap$, $\in$, $\notin$, $\subset$, $\subseteq$, $\not\subseteq$, $=$, and $\neq$ with sets
$\times$ for the Cartesian product of sets
$\rightarrow$ for a mapping between sets
$|$ for the divides relation between integer values
x^y, $\sqrt{x}$, $\log_b(x)$ on real values