INTERNATIONAL STANDARD

ISO/IEC 1539

Second edition
1991-07-01

Information technology — Programming languages — Fortran

Technologies de l'information — Langages de programmation — Fortran

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 1539:1991
https://standards.iteh.ai/catalog/standards/sist/58d391bf-e571-4a5e-91de-4ad20949d95b/iso-iec-1539-1991

Reference number
ISO/IEC 1539 : 1991 (E)

Contents

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 1539:1991
https://standards.iteh.ai/catalog/standards/sist/58d391bf-e571-4a5e-91de-4ad20949d95b/iso-iec-1539-1991

© ISO/IEC 1991
All rights reserved. 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 the publisher.

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland
Printed in Switzerland

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 1539:1991
https://standards.iteh.ai/catalog/standards/sist/58d391bf-e571-4a5e-91de-4ed20949d95b/iso-iec-1539-1991

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 1539:1991
https://standards.iteh.ai/catalog/standards/sist/58d391bf-e571-4a5e-91de-4a20949d95b/iso-iec-1539-1991

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 1539:1991
https://standards.iteh.ai/catalog/standards/sist/58d391bf-e571-4a5e-91de-
ad9049d05b/iso-iec-1539-1991

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 1539:1991
https://standards.iteh.ai/catalog/standards/sist/58d3916f-e571-4a5e-91de-4ad20949d95b/iso-iec-1539-1991

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 1539:1991
https://standards.iteh.ai/catalog/standards/sist/58d391bf-e571-4a5e-91de-4ad20949d95b/iso-iec-1539-1991

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 1539:1991
https://standards.iteh.ai/catalog/standards/sist/58d3911b-e571-4a5e-91de-4ad20949d95b/iso-iec-1539-1991

Annexes

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 1539:1991
https://standards.iteh.ai/catalog/standards/sist/38d391bf-e571-4a5e-91de-4ad2091fd938/iso-iec-1539-1991

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 1539:1991
https://standards.iteh.ai/catalog/standards/sist/58d391bf-e571-4a5e-91de-4ad20949d95b/iso-iec-1539-1991

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 1539:1991
https://standards.iteh.ai/catalog/standards/sist/58d391bf-e571-4a5e-91de-4ad20949d95b/iso-iec-1539-1991

Figures

Tables

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 1539 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

This second edition cancels and replaces the first edition (ISO 1539 : 1980), which has been technically revised.

Annexes A, B, C, D, E and F are for information only.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 1539:1991
https://standards.iteh.ai/catalog/standards/sist/58d391bf-e571-4a5e-91de-4ad20949d95b/iso-iec-1539-1991

Introduction

Standard programming language Fortran

This International Standard specifies the form and establishes the interpretation of programs expressed in the Fortran language (known informally as "Fortran 90"). It consists of the specification of the language Fortran. No subsets are specified in this International Standard. With limitations noted in 1.4.1, the syntax and semantics of the International Standard commonly known as "FORTRAN 77" are contained entirely within this International Standard. Therefore, any standard-conforming FORTRAN 77 program is standard conforming under this International Standard. New features can be compatibly incorporated into such programs, with any exceptions indicated in the text of this International Standard.

A standard-conforming Fortran processor is also a standard-conforming FORTRAN 77 processor.

Note that the name of this language, Fortran, differs from that in FORTRAN 77 in that only the first letter is capitalized. Both FORTRAN 77 and FORTRAN 66 used only capital letters in the official name of the language, but Fortran 90 does not continue this tradition.

Overview

Among the additions to FORTRAN 77 in this International Standard, seven stand out as the major ones:

(1) Array operations

iTeh STANDARD PREVIEW

(2) Improved facilities for numerical computation

(standards.iteh.ai)

(3) Parameterized intrinsic data types

(4) User-defined data types

ISO/IEC 1539:1991
https://standards.iteh.ai/catalog/standards/sist/58d391bf-e571-4a5e-91de-4ad2094 9d950/iso-iec-1539-1991

(5) Facilities for modular data and procedure definitions

(6) Pointers

(7) The concept of language evolution

A number of other additions are also included in this International Standard, such as improved source form facilities, more control constructs, recursion, additional input/output facilities, and dynamically allocatable arrays.

Array operations

Computation involving large arrays is an important part of engineering and scientific computing. Arrays may be used as entities in Fortran. Operations for processing whole arrays and subarrays (array sections) are included in the language for two principal reasons: (1) these features provide a more concise and higher level language that will allow programmers more quickly and reliably to develop and maintain scientific/engineering applications, and (2) these features can significantly facilitate optimization of array operations on many computer architectures.

The FORTRAN 77 arithmetic, logical, and character operations and intrinsic (predefined) functions are extended to operate on array-valued operands. The array extensions include whole, partial, and masked array assignment, array-valued constants and expressions, and facilities to define user-supplied array-valued functions. New intrinsic procedures are provided to manipulate and construct arrays, to perform gather/scatter operations, and to support extended computational capabilities involving arrays. For example, an intrinsic function is provided to sum the elements of an array.

Numerical computation

Scientific computation is one of the principal application domains of Fortran, and a guiding objective for all of the technical work is to strengthen Fortran as a vehicle for implementing scientific software. Though nonnumeric computations are increasing dramatically in scientific applications, numeric computation remains dominant. Accordingly, the additions include portable control over numeric precision specification, inquiry as to the characteristics of numeric representation, and improved control of the performance of numerical programs (for example, improved argument range reduction and scaling).

Parameterized character data type

Optional facilities for multibyte character data for languages with large character sets, such as those in China and Japan, are added by using a kind parameter for the character data type. This facility allows additional character sets for special purposes as well, such as characters for mathematics, chemistry, or music.

Derived types

"Derived type" is the term given to that set of features in this International Standard that allows the programmer to define arbitrary data structures and operations on them. Data structures are user-defined aggregations of intrinsic and derived data types. Intrinsic uses of structured objects include assignment, input/output, and as procedure arguments. With no additional derived-type operations defined by the user, the derived data type facility is a simple data structuring mechanism. With additional operation definitions, derived types provide an effective implementation mechanism for data abstractions.

Procedure definitions may be used to define operations on intrinsic or derived types and nonintrinsic assignments for intrinsic and derived types.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

Modular definitions

In FORTRAN 77, there was no way to define a global data area in only one place and have all the program units in an application use that definition. In addition, the ENTRY statement is awkward and restrictive for implementing a related set of procedures, possibly involving common data objects. Finally, there was no means in FORTRAN 77 by which procedure definitions, especially interface information, could be made known locally to a program unit. These and other deficiencies are remedied by a new type of program unit that may contain any combination of data object declarations, derived-type definitions, procedure definitions, and procedure interface information. This program unit, called a module, may be considered to be a generalization and replacement for the block data program unit. A module may be accessed by any program unit, thereby making the module contents available to that program unit. Thus, modules provide improved facilities for defining global data areas, procedure packages, and encapsulated data abstractions.

Pointers

Pointers allow arrays to be sized dynamically and ranged, and structures to be linked to create lists, trees, and graphs. An object of any intrinsic or derived type may be declared to have the pointer attribute. Once such an object becomes associated with a target, it may appear almost anywhere a nonpointer object with the same type, type parameters, and shape may appear.

Language evolution

With the addition of new facilities, certain old features become redundant and may eventually be phased out of the language as their usage declines. For example, the numeric facilities alluded to above provide the functionality of double precision; with the new array facilities, nonconformable argument association (such as associating an array element with a dummy array) is unnecessary (and in fact is not useful as an array operation); and block data program units are redundant and inferior to modules.

As part of the evolution of the language, categories of language features (deleted and obsolescent) are provided which allow unused features of the language to be removed from future standards.

Organization of this International Standard

This document is organized in 14 sections, dealing with 7 conceptual areas. These 7 areas, and the sections in which they are treated, are:

High/Low Level Concepts	Sections 1, 2, 3
Data Concepts	Sections 4, 5, 6
Computations	Sections 7, 13
Execution Control	Section 8
Input/Output	Sections 9, 10
Program Units	Sections 11, 12
Scoping and Association Rules	Section 14

High/low level concepts

Section 2 (Fortran Terms and Concepts) contains many of the high level concepts of Fortran. This includes the concept of an executable program and the relationships among its major parts. Also included are the syntax of program units, the rules for statement ordering, and the definitions of many of the fundamental terms used throughout the document.

Section 3 (Characters, Lexical Tokens, and Source Form) describes the low level elements of Fortran, such as the character set and the allowable forms for source programs. It also contains the rules for constructing literal constants and names for Fortran entities, and lists all of the Fortran operators.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

Data concepts

The array operations (arrays as data objects) and data structures provide a rich set of data concepts in Fortran. The main concepts are those of data type, data object, and the use of data objects, which are described in Sections 4, 5, and 6, respectively.

Section 4 (Intrinsic and Derived Data Types) describes the distinction between a data type and a data object, and then focuses on data type. It defines a data type as a set of data values, corresponding forms (constants) for representing these values, and operations on these values. The concept of an intrinsic data type is introduced, and the properties of Fortran's intrinsic types (INTEGER, REAL, COMPLEX, LOGICAL, and CHARACTER) are described. Note that only type concepts are described here, and not the declaration and properties of data objects.

Section 4 also introduces the concept of derived (user-defined) data types, which are compound types whose components ultimately resolve into intrinsic types. The details of defining a derived type are given (note that this has no counterpart with intrinsic types as intrinsic types are predefined and therefore need not—indeed cannot—be redefined by the programmer). As with intrinsic types, this section deals only with type properties, and not with the declaration of data objects of derived type.

Section 5 (Data Object Declarations and Specifications) describes in detail how named data objects are declared and given the desired properties (attributes). An important attribute (the only one required for each data object) is the object's data type, so the type declaration statement is the main feature of this section. The various attributes are described in detail, as well as the two ways that attributes may be specified (type declaration statements and attribute specification statements). Implicit typing and storage association (COMMON and EQUIVALENCE) are also described in this section, as well as data object value initialization.

Section 6 (Use of Data Objects) deals mainly with the concept of a variable, and describes the various forms that variables may take. Scalar variables include character strings and substrings, structured (derived-type) objects, structure components, and array elements. Arrays are considered to be variables, as are array sections. Among the array facilities described here are array sections (subarrays), and array