

	Informationsverarbeitung Programmiersprache COBOL Modul für trigonometrische Funktionen (ISO 1989 : 1985/Amd. 1 : 1992) Englische Fassung EN 21989 : 1989/A1 : 1993 Änderung A1	DIN 66028/A1
Diese Norm enthält die Internationale Norm ISO 1989 Amd. 1 ICS 35.060 Änderung von DIN 66028 : 1986-08 Deskriptoren: Informationsverarbeitung, Programmiersprache, COBOL Information processing – Programming language COBOL – Intrinsic function module (ISO 1989 : 1985/Amd. 1 : 1992) English version EN 21989 : 1989/A1 : 1993 Amendment A1 Traitement de l'information – Langage de programmation COBOL – Module de fonction intrinsèque (ISO 1989 : 1985/Amd. 1 : 1992) Version anglaise EN 21989 : 1989/A1 : 1993 Amendement A1 Die Europäische Norm EN 21989 : 1989/A1 : 1993 hat den Status einer Deutschen Norm. Nationales Vorwort Die vorliegende Änderung A1 ergänzt die gültige Norm über die Programmiersprache COBOL, siehe DIN 66028 : 1986-08, um ein Modul, das z. B. trigonometrische Funktionen, Logarithmen, arithmetische Mittel und Standardabweichungen enthält. Dieses Modul ermöglicht, Datenelemente zu verwenden, deren Wert maschinell während der Ausführung eines Programms berechnet wird. Internationale Patentklassifikation G 06 F 009/45 Fortsetzung 2 Seiten EN sowie 78 Seiten ISO-Original Normenausschuß Informationsverarbeitungssysteme (NI) im DIN Deutsches Institut für Normung e.V.		

DK 681.3.06 : 800.92

Deskriptoren: Anforderung – COBOL – Datenverarbeitung – Informationsverarbeitung – Programmiersprache

Deutsche Fassung

Programmiersprachen

COBOL

Modul für trigonometrische Funktionen
(ISO 1989 : 1985, Änderung 1 : 1992)

Programming languages – COBOL – Intrinsic
function module
(ISO 1989 : 1985, Amendment 1 : 1992)

Langages de programmation – COBOL –
Module de fonction intrinsèque
(ISO 1989 : 1985, Amendement 1 : 1992)

Diese Änderung 1 modifiziert die Europäische Norm EN 21989 : 1989.

Diese Änderung wurde von CEN am 1993-11-25 angenommen.

Die CEN-Mitglieder sind gehalten, die CEN/CENELEC-Geschäftsordnung zu erfüllen, in der die Bedingungen festgelegt sind, unter denen dieser Europäischen Norm ohne jede Änderung der Status einer nationalen Norm zu geben ist.

Auf dem letzten Stand befindliche Listen dieser nationalen Normen mit ihren bibliographischen Angaben sind beim Zentralsekretariat oder bei jedem CEN-Mitglied auf Anfrage erhältlich.

Diese Europäische Norm besteht in drei offiziellen Fassungen (Deutsch, Englisch, Französisch). Eine Fassung in einer anderen Sprache, die von einem CEN-Mitglied in eigener Verantwortung durch Übersetzung in seine Landessprache gemacht und dem Zentralsekretariat mitgeteilt worden ist, hat den gleichen Status wie die offiziellen Fassungen.

CEN-Mitglieder sind die nationalen Normungsinstitute von Belgien, Dänemark, Deutschland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Luxemburg, Niederlande, Norwegen, Österreich, Portugal, Schweden, Schweiz, Spanien und dem Vereinigten Königreich.

CEN

EUROPÄISCHES KOMITEE FÜR NORMUNG
European Committee for Standardization
Comité Européen de Normalisation

Zentralsekretariat: rue de Stassart 36, B-1050 Brüssel

© 1993. Das Copyright ist den CEN-Mitgliedern vorbehalten.

Ref.-Nr. EN 21989 : 1989/A1 : 1993 D

Vorwort

Das Technische Büro hatte entschieden, die Europäische Änderung:

“Programmiersprachen – COBOL – Modul für trigonometrische Funktionen (ISO 1989 : 1985, Änderung 1 : 1992)”

hiermit zum Einstufigen Annahmeverfahren vorzulegen.

Das Ergebnis der Einstufigen Annahmeverfahren war positiv.

Zur Zeit besteht dieses Dokument nur in Englisch.

Diese Europäische Norm muß den Status einer nationalen Norm enthalten, entweder durch Veröffentlichung eines identischen Textes oder durch Anerkennung bis Mai 1994, und etwaige entgegenstehende nationale Normen müssen bis Mai 1994 zurückgezogen werden.

Entsprechend der CEN/CENELEC-Geschäftsordnung sind folgende Länder gehalten, diese Europäische Norm zu übernehmen:

Belgien, Dänemark, Deutschland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Luxemburg, Niederlande, Norwegen, Österreich, Portugal, Schweden, Schweiz, Spanien und das Vereinigte Königreich.

Anerkennungsnotiz

Der Text der Internationalen Norm ISO 1989 : 1985, Änderung 1 : 1992 wurde von CEN ohne jegliche Änderung als Europäische Änderung angenommen.

TABLE OF CONTENTS

ADDENDUM 1: CHANGES TO STANDARD COBOL	A-1
--	------------

SECTION A: INTRINSIC FUNCTION MODULE

Chapter 1: Introduction to the Intrinsic Function Module

1.1 Purpose of Intrinsic Function Module	A-27
1.2 Language Concepts	A-27

Chapter 2: General Description

2.1 Function Definition and Returned Value	A-27
2.2 Arguments	A-28
2.3 Types of Functions	A-29
2.4 Definitions of Functions	A-29
2.5 The ACOS Function	A-33
2.6 The ANNUITY Function	A-34
2.7 The ASIN Function	A-35
2.8 The ATAN Function	A-36
2.9 The CHAR Function	A-37
2.10 The COS Function	A-38
2.11 The CURRENT-DATE Function	A-39
2.12 The DATE-OF-INTEGGER Function	A-41
2.13 The DAY-OF-INTEGGER Function	A-42
2.14 The FACTORIAL Function	A-43
2.15 The INTEGER Function	A-44
2.16 The INTEGER-OF-DATE Function	A-45
2.17 The INTEGER-OF-DAY Function	A-46
2.18 The INTEGER-PART Function	A-47
2.19 The LENGTH Function	A-48
2.20 The LOG Function	A-49
2.21 The LOG10 Function	A-50
2.22 The LOWER-CASE Function	A-51
2.23 The MAX Function	A-52
2.24 The MEAN Function	A-53
2.25 The MEDIAN Function	A-54
2.26 The MIDRANGE Function	A-55
2.27 The MIN Function	A-56
2.28 The MOD Function	A-57
2.29 The NUMVAL Function	A-58
2.30 The NUMVAL-C Function	A-59
2.31 The ORD Function	A-60
2.32 The ORD-MAX Function	A-61
2.33 The ORD-MIN Function	A-62
2.34 The PRESENT-VALUE Function	A-63
2.35 The RANDOM Function	A-64
2.36 The RANGE Function	A-65
2.37 The REM Function	A-66
2.38 The REVERSE Function	A-67

2.39	The SIN Function	A-68
2.40	The SQRT Function	A-69
2.41	The STANDARD-DEVIATION Function.	A-70
2.42	The SUM Function	A-71
2.43	The TAN Function	A-72
2.44	The UPPER-CASE Function	A-73
2.45	The VARIANCE Function	A-74
2.46	The WHEN-COMPILED Function.	A-75

Programming Language – Intrinsic Function Module for COBOL

ADDENDUM 1: CHANGES TO STANDARD COBOL

The following are the changes to be applied to the document entitled "American National Standard for Information Systems - Programming Language - COBOL, ANSI X3.23-1985, ISO 1989-1985", in order to include the language elements of the Intrinsic Function module and to correct typographical errors (indicated by the symbols (T) in the page number column). The changes add totally upward compatible language extensions to Standard COBOL except for the addition of the reserved word FUNCTION.

Page No.	Change to ANSI X3.23-1985 and ISO 1989-1985
ii	Add the following after the entry for section II, paragraph 7: 8. Intrinsic Function Facility II-36
iii	Change the entry for section VI, paragraph 1.3 to read: 1.3 Restrictions on Overall Language. VI-1
xi	Add the following after the entry for 3.4: 3.5 American National Standard COBOL 1985, Addendum 1 XVII-13
xi	Add the following after the entry for 4.3: 4.4 ISO Standard 1989-1985 for COBOL, Addendum 1 XVII-15
I-1	Paragraph 1.2, first paragraph, second sentence, change to read: "The standard defines 12 functional processing modules: Nucleus, Sequential I-O, Relative I-O, Indexed I-O, Inter-Program Communication, Sort-Merge, Source Text Manipulation, Report Writer, Communication, Debug, Segmentation, and Intrinsic Function."
I-1	Paragraph 1.2, first paragraph, last sentence, change to read: "Three of the modules contain only level 1 elements."
I-3	Paragraph 1.2, add the following paragraph after the last paragraph: The Intrinsic Function module provides the capability to reference a data item whose value is derived automatically at the time of reference during the execution of the object program. The Intrinsic Function module contains only level 1 elements.

- I-3 Paragraph 1.3, first paragraph, first sentence, change to read: "...into nineteen sections."
- I-3 Paragraph 1.3, third paragraph, first sentence, change to read: "Sections VI through XVI and section A contain specifications for the twelve functional...".
- I-3 Paragraph 1.3, fourth paragraph, change to read: "Sections II through XVI and section A comprise the detailed...".
- I-4 Paragraph 1.4, fourth paragraph, first sentence, change to read: "...representation of the 12 functional processing modules...".
- I-4 Paragraph 1.4, fifth paragraph, add to the list:
- ITR Intrinsic Function
- I-5 Change diagram to include the Intrinsic Function module as shown on the next page of this addendum.

REQUIRED MODULES (Required in Subsets)							OPTIONAL MODULES (Not Required in Subsets)								
Nucleus	Sequential I/O	Relative I/O	Indexed I/O	Inter-Program Communication	Sort-Merge	Source Text Manipulation	Report Writer	Communication	Debug	Segmentation	Intrinsic Function				
2 NUC 1,2	2 SEQ 1,2	2 REL 0,2	2 INX 0,2	2 IPC 1,2	1 SRT 0,1	2 STM 0,2	1 RPW 0,1	2 COM 0,2	2 DEB 0,2	2 SEG 0,2	1 ITR 0,1				
1 NUC 1,2	1 SEQ 1,2	1 REL 0,2	1 INX 0,2	1 IPC 1,2	1 SRT 0,1	1 STM 0,2		1 COM 0,2	1 DEB 0,2	1 SEG 0,2					
1 NUC 1,2	1 SEQ 1,2	NULL	NULL	1 IPC 1,2	NULL	NULL									
H I G H															
I N T E R M E D I A T E															
M I N I M U M															
C O B O L S U B S E T S															

- I-6 Paragraph 1.5, second sentence, change "consists of 11 modules, seven of which are required and four of which are optional." to "consists of 12 modules, seven of which are required and five of which are optional. (It is the intention of X3J4 to require the Intrinsic Function module in the next complete revision of Standard COBOL.)"
- I-6 Paragraph 1.5.1, first paragraph, last sentence, change to read: "The five optional modules (Report Writer, Communication, Debug, Segmentation, and Intrinsic Function) are not..."
- I-8 Paragraph 1.5.2.5.3, change to read:
- 1.5.2.5.3 Reserved Words**
- An implementation of Standard COBOL must recognize as reserved words all of the COBOL reserved words occurring in the specification of the seven required modules and the four optional modules of Report Writer, Communication, Debug, and Segmentation. An implementation of Standard COBOL need not recognize any new reserved words introduced by the optional Intrinsic Function module until that module is included in the implementation. (See page IV-45, COBOL Reserved Words.)
- I-10 Paragraph 2.1, last paragraph, add the following to the list:
- Page I-39: Summary of elements in the Intrinsic Function module
- I-12 Change the entry for "Reference modification" to "Reference-modifier".
- I-12 (T) Under "Reference Format", change "Asterisk (8) comment line" to "Asterisk (*) comment line".
- I-39 After the Segmentation module list, add the Intrinsic Function module list shown on the next page of this addendum.

SUMMARY OF ELEMENTS IN THE INTRINSIC FUNCTION MODULE

ELEMENT	LEVEL 1
<u>LANGUAGE CONCEPTS</u>	
<u>Character-Strings</u>	
COBOL words	
Function-name	X
<u>Uniqueness of Reference</u>	
Function-Identifier	X
<u>PROCEDURE DIVISION</u>	
ACOS function	X
ANNUITY function	X
ASIN function	X
ATAN function	X
CHAR function	X
COS function	X
CURRENT-DATE function	X
DATE-OF-INTEGER function	X
DAY-OF-INTEGER function	X
FACTORIAL function	X
INTEGER function	X
INTEGER-OF-DATE function	X
INTEGER-OF-DAY function	X
INTEGER-PART function	X
LENGTH function	X
LOG function	X
LOG10 function	X
LOWER-CASE function	X
MAX function	X
MEAN function	X
MEDIAN function	X
MIDRANGE function	X
MIN function	X
MOD function	X
NUMVAL function	X
NUMVAL-C function	X
ORD function	X
ORD-MAX function	X
ORD-MIN function	X
PRESENT-VALUE function	X
RANDOM function	X
RANGE function	X
REM function	X
REVERSE function	X
SIN function	X
SQRT function	X
STANDARD-DEVIATION function	X
SUM function	X
TAN function	X
UPPER-CASE function	X
VARIANCE function	X
WHEN-COMPILED function	X

- I-40 Paragraph 3.1, second paragraph, add the following to the list:
- ITR Intrinsic Function
- I-41 (T) Under "Character Set", change entry in MODULE column for "Characters used in punctuation =" from "1 STM" to "2 STM".
- I-42 Add the following before the entry for "Literals"; align the word "Function-names" one position to the right of the word "Literals":
- Function-names 1 ITR
- I-42 Change the entry for "Reference modification" to "Reference-modifier".
- I-42 Add the following before the entry for "Reference-modifier"; align the word "Function-identifier" with the word "Reference-modifier":
- Function-identifier 1 ITR
- I-45 (T) Under "OBJECT-COMPUTER paragraph", change entry in MODULE column for "SEGMENT-LIMIT clause" from "1 SEG Z" to "2 SEG Z".
- I-51 (T) In the line following "Level-number clause", change "may be 1 or 1 digits" to "may be 1 or 2 digits".
- I-54 Add the following before the entry for "ACCEPT statement":
- Intrinsic functions
- | | |
|------------------------------------|-------|
| ACOS function | 1 ITR |
| ANNUITY function | 1 ITR |
| ASIN function | 1 ITR |
| ATAN function | 1 ITR |
| CHAR function | 1 ITR |
| COS function | 1 ITR |
| CURRENT-DATE function | 1 ITR |
| DATE-OF-INTEGER function | 1 ITR |
| DAY-OF-INTEGER function | 1 ITR |
| FACTORIAL function | 1 ITR |
| INTEGER function | 1 ITR |
| INTEGER-OF-DATE function | 1 ITR |
| INTEGER-OF-DAY function | 1 ITR |
| INTEGER-PART function | 1 ITR |
| LENGTH function | 1 ITR |
| LOG function | 1 ITR |
| LOG10 function | 1 ITR |
| LOWER-CASE function | 1 ITR |
| MAX function | 1 ITR |
| MEAN function | 1 ITR |
| MEDIAN function | 1 ITR |
| MIDRANGE function | 1 ITR |
| MIN function | 1 ITR |
| MOD function | 1 ITR |

NUMVAL function	1 ITR
NUMVAL-C function	1 ITR
ORD function	1 ITR
ORD-MAX function	1 ITR
ORD-MIN function	1 ITR
PRESENT-VALUE function	1 ITR
RANDOM function	1 ITR
RANGE function	1 ITR
REM function	1 ITR
REVERSE function	1 ITR
SIN function	1 ITR
SQRT function	1 ITR
STANDARD-DEVIATION function	1 ITR
SUM function	1 ITR
TAN function	1 ITR
UPPER-CASE function	1 ITR
VARIANCE function	1 ITR
WHEN-COMPILED function	1 ITR

Statements

- I-54 thru I-63 Indent all lines following the newly inserted "Statements" entry beginning with "ACCEPT statement" and ending with "END-WRITE phrase".
- II-23 (T) Paragraph 6.4.1.1, entitled "Names of Programs", second paragraph, first line, change "compiled program" to "compiled programs".
- II-25 (T) Paragraph 6.4.2.2, entitled "Value of Parameters", second paragraph, penultimate line, change "may be used by a called program to return to the" to "may be used by a called program to return a result to the".
- II-35 Add the following on page II-36 after paragraph 7.6:

8. INTRINSIC FUNCTION FACILITY

Data processing problems frequently require the use of values which are not directly accessible in the data storage associated with the object program. These data values must be derived through performing some operations on other data. A function represents a temporary data item whose value is derived automatically at the time of reference during the execution of the object program.

The value returned by a function is considered to be a data value. A mechanism is provided at object time to assign a data value to a function when it is referenced. In order to determine the value of a function, the evaluation mechanism may require access to data values provided by the referencing program. These data values are provided by specifying parameters, known as arguments, when referencing the function. Specific functions may place constraints on these arguments such as range, data types, or size, etc. If, at the time a function is referenced, the arguments specified for that reference do not have values that comply with the specified constraints, the returned value for the function is undefined.

III-1
thru
III-26

Insert the following terms into the glossary at the appropriate position in the alphabetic sequence:

Alphanumeric Function. A function whose value is composed of a string of one or more characters from the computer's character set.

Argument. An identifier, a literal, or an arithmetic expression that specifies a value to be used in the evaluation of a function.

Function. A temporary data item whose value is determined by invoking a mechanism provided by the implementor at the time the function is referenced during the execution of a statement.

Function-Identifier. A syntactically correct combination of character-strings and separators that references a function. The data item represented by a function is uniquely identified by a function-name with its arguments, if any. A function-identifier may include a reference-modifier. A function-identifier that references an alphanumeric function may be specified anywhere in the general formats that an identifier may be specified, subject to certain restrictions. A function-identifier that references an integer or numeric function may be referenced anywhere in the general formats that an arithmetic expression may be specified. (See page IV-22, Function-Identifier.)

Function-Name. A word that names a mechanism provided by the implementor to determine the value of a function.

Integer Function. A function whose category is numeric and whose definition provides that all digits to the right of the decimal point are zero in the returned value for any possible evaluation of the function.

Numeric Function. A function whose class and category are numeric but which for some possible evaluation does not satisfy the requirements of an integer function.

III-1
thru
III-26

Replace the definitions as follows:

Data Item. A unit of data (excluding literals) defined by a COBOL program or by the rules for function evaluation.

Identifier. A syntactically correct combination of character-strings and separators that names a data item. When referencing a data item which is not a function, an identifier consists of a data-name, together with its qualifiers, subscripts, and reference-modifier, as required for uniqueness of reference. When referencing a data item which is a function, a function-identifier is used. The rules for 'identifier' associated with general formats may, however, specifically prohibit reference to functions, qualification, subscripting, or reference modification.

Integer. (1) A numeric literal that does not include any digit positions to the right of the decimal point.

(2) A numeric data item defined in the Data Division that does not include any digit positions to the right of the decimal point.

(3) A numeric function whose definition provides that all digits to the right of the decimal point are zero in the returned value for any possible evaluation of the function.

Where the term 'integer' appears in the general formats, integer must be a numeric literal which is an integer, and it must be neither signed nor zero unless explicitly allowed by the rules for that format.

Key Word. A reserved word or function-name whose presence is required when the format in which the word appears is used in a source program.

Subscript. An occurrence number represented by either an integer, a data-name optionally followed by an integer with the operator + or -, or an index-name optionally followed by an integer with the operator + or -, that identifies a particular element in a table. A subscript may be the word ALL when the subscripted identifier is used as a function argument. (See page A-28, Arguments.)

Word. A character-string of not more than 30 characters which forms a user-defined word, a system-name, a reserved word, or a function-name. (See page IV-5, COBOL Words.)

III-19 Replace the definition for Reference Modifier with the following:

Reference-Modifier. A syntactically correct combination of character-strings and separators that defines a unique data item. It includes a delimiting left parenthesis separator, the leftmost character position, a colon separator, optionally a length, and a delimiting right parenthesis separator. (See page IV-22, Reference-Modifier.)

IV-2 (T) Paragraph 2.1.5 entitled "Ellipses", second paragraph, first line, change "In the general format," to "In the general formats,".

IV-4 (T) Paragraph 4.1, entitled "Character Set", third paragraph, first line, change "fewer than 51 characters, double" to "fewer than 52 characters (all characters of the COBOL character set except the lowercase letters), double".

IV-4 Paragraph 4.2.1, rule (4), change second sentence to read: "Parentheses may appear only in balanced pairs of left and right parentheses delimiting subscripts, a list of function arguments, reference modifiers, arithmetic expressions, or conditions."

IV-5 Paragraph 4.2.2.1, replace with the following:

A COBOL word is a character-string of not more than 30 characters which forms a user-defined word, a system-name, a reserved word, or a function-name. Each character of a COBOL word is selected from the set of letters, digits, and the hyphen. The hyphen may not appear as the first or last character. Each lowercase letter is considered to be equivalent to its corresponding uppercase letter. Within a source program, reserved words and user-defined words form disjoint sets; reserved words and system-names form disjoint sets; reserved words and function-names form disjoint sets; function-names, system-names, and user-defined words form intersecting sets. The same COBOL word may be used as a function-name, as a system-name, and as a user-defined word within a source program; and the class of a specific occurrence of the COBOL word is determined by the context of the clause or phrase in which it occurs.

IV-9 Add the following before paragraph 4.2.2.2:

4.2.2.1.4 Function-Names

A function-name is a word that is one of a specified list of words which may be used in COBOL source programs. The same word, in a different context, may appear in a program as a user-defined word or a system-name. (See page A-29, Definitions of Functions.)

IV-15 Paragraph 4.3.3, delete the last sentence and add the following new paragraphs:

Every data item which is a function is an elementary item, and belongs to one of the categories alphanumeric or numeric, and to the corresponding class; the category of each function is determined by the definition of the function. This definition is made in these specifications. (See page A-27, Intrinsic Function Module.)

The following table depicts the relationship of the class and categories of data items.

IV-16 Paragraph 4.3.4, second paragraph, replace first sentence with the following: "When a computer provides more than one means of representing data, the standard data format must be used for data items other than integer and numeric functions, if not otherwise specified by the data description."

IV-16 Paragraph 4.3.4, add the following new paragraph preceding the last paragraph:

An alphanumeric function is always represented in the standard data format. The size of an alphanumeric function in standard data format characters is determined by the definition of the function. The implementor specifies the representation of integer and numeric functions, and this representation need not be the standard data format. Integer and numeric functions may be used only in arithmetic expressions. An integer or numeric function represents the value resulting from the evaluation of the function without the restriction on composite of operands and/or receiving data items. Each implementor will indicate the techniques used in evaluating integer and numeric functions.

IV-18 Paragraph 4.3.8, add the following new paragraph preceding the last paragraph:

When the resource named is a function, the function definition may require the user to specify in the reference to the function a value or set of values for one or more parameters which determine the value of the function for that particular reference. This is accomplished through the specification of arguments as described in the following paragraphs.

IV-21 Paragraph 4.3.8.2.2, replace the general format with the following:

$$\left\{ \begin{array}{l} \text{condition-name-1} \\ \text{data-name-1} \end{array} \right\} \left(\begin{array}{l} \text{ALL} \\ \text{integer-1} \\ \text{data-name-2} \left[\begin{array}{l} + \\ - \end{array} \right] \text{integer-2} \\ \text{index-name-1} \left[\begin{array}{l} + \\ - \end{array} \right] \text{integer-3} \end{array} \right) \dots$$

IV-21 Paragraph 4.3.8.2.3, add new syntax rule 7 as follows:

(7) The subscript ALL may be used only when the subscripted identifier is used as a function argument and may not be used when condition-name-1 is specified. (See page A-28, Arguments.)

IV-22 Renumber paragraphs 4.3.8.3 thru 4.3.8.5 to 4.3.8.4 thru 4.3.8.6, respectively.
thru
IV-24

IV-22 Add the following as new paragraph 4.3.8.3:

4.3.8.3 Function-Identifier

4.3.8.3.1 Purpose of a Function-Identifier

A function-identifier is a syntactically correct combination of character-strings and separators that uniquely references the data item resulting from the evaluation of a function.

4.3.8.3.2 General Format

FUNCTION function-name-1 [({argument-1} ...)] [reference-modifier]

4.3.8.3.3 Syntax Rules

(1) Argument-1 must be an identifier, a literal, or an arithmetic expression. Specific rules governing the number, class, and category of argument-1 are given in the definition of each function. (See page A-27, Intrinsic Function Module.)

(2) A reference-modifier may be specified only for functions of the category alphanumeric.

(3) A function-identifier which references an alphanumeric function may be specified anywhere in the general formats that an identifier is permitted and where the rules associated with the general formats do not specifically prohibit reference to functions, except as follows:

a. As a receiving operand of any statement.

b. Where the rules associated with the general formats require the data item being referenced to have particular characteristics (such as class and category, usage, size, sign, and permissible values) and the evaluation of the function according to its definition and the particular arguments specified would not have these characteristics.

(4) A function-identifier which references an integer or numeric function may be used only in an arithmetic expression.

4.3.8.3.4 General Rules

(1) The class and other characteristics of the function being referenced are determined by the function definition. (See page A-27, Intrinsic Functions.)

(2) At the time reference is made to a function, its arguments are evaluated individually in the order specified in the list of arguments, from left to right. An argument being evaluated may itself be a function-identifier or may be an expression containing function-identifiers. There is no restriction preventing the function referenced in evaluating an argument from being the same function as that for which the argument is specified.

IV-22 New paragraph 4.3.8.4, change the title to "Reference-Modifier".

IV-22 New paragraph 4.3.8.4.2, replace the general format with the following:

```
{ data-name-1
  FUNCTION function-name-1 [((argument-1) ... )] } (leftmost-character-position:[length])
```

NOTE: Data-name-1 and FUNCTION function-name-1 (argument-1) are shown in the above format to provide context and are not part of the reference-modifier.

IV-22 New paragraph 4.3.8.4.3, add the following as new syntax rule 5:

(5) The function referenced by function-name-1 and its arguments, if any, must be an alphanumeric function.

IV-22 New paragraph 4.3.8.4.4, change the first sentence of general rule 1 to read: "Each character of a data item referenced by data-name-1 or by function-name-1 and its arguments, if any, is assigned an ordinal number incrementing by one from the leftmost position to the rightmost position."

IV-23 New paragraph 4.3.8.4.4, general rule 3a, add the following new last sentence: "If an ALL subscript is specified for an operand, the reference-modifier is applied to each of the implicitly specified elements of the table."

IV-23 New paragraph 4.3.8.4.4, general rule 3, add the following as new rule c:

c. If reference modification is specified in a function reference, the reference modification is evaluated immediately after evaluation of the function.

- IV-23 New paragraph 4.3.8.4.4, general rule 4, change the first sentence to read: "Reference modification creates a unique data item that is a subset of the data item referenced by data-name-1 or by function-name-1 and its arguments, if any."
- IV-23 New paragraph 4.3.8.4.4, general rule 4, paragraphs a and b, replace four occurrences of "the data item referenced by data-name-1" by "the data item referenced by data-name-1 or function-name-1 and its arguments, if any".
- IV-23 New paragraph 4.3.8.4.4, general rule 5, second sentence, replace the word "It" by "When a function is referenced, the unique data item has the class and category of alphanumeric. When data-name-1 is specified, the unique data item".
- IV-23 New paragraph 4.3.8.5, change to read:

4.3.8.5 Identifier

4.3.8.5.1 Function

An identifier is a syntactically correct sequence of character-strings and separators used to reference data uniquely.

When a data item other than a function is being referenced, identifier is a term used to reflect that a data-name, if not unique in a program, must be followed by a syntactically correct combination of qualifiers, subscripts, or reference modifiers necessary for uniqueness of reference. (See page X-4, Scope of Names.)

4.3.8.5.2 General Format

Format 1:

function-identifier-1

Format 2:

$$\text{data-name-1} \left[\begin{array}{c} \text{IN} \\ \text{OF} \end{array} \right] \text{data-name-2} \dots \left[\begin{array}{c} \text{IN} \\ \text{OF} \end{array} \right] \left\{ \begin{array}{l} \text{cd-name-1} \\ \text{file-name-1} \\ \text{report-name-1} \end{array} \right\}$$

[({subscript} ...)] [reference-modifier]

4.3.8.5.3 Syntax Rules

- (1) The words IN and OF are equivalent.

- IV-45 Paragraph 8, insert the following before the list of reserved words:

The following is a list of COBOL reserved words for the seven required modules and the four optional modules of Report Writer, Communication, Debug, and Segmentation.

IV-46 Paragraph 8, insert the following after the list of reserved words:

The following is a list of COBOL reserved words for the optional Intrinsic Function module.

FUNCTION

- V-3 (T) SYMBOLIC CHARACTERS clause, delete outermost set of braces.
- V-7 (T) ORGANIZATION clause, delete second occurrence of a right bracket after the word SEQUENTIAL.
- V-15 (T) Insert a terminal period following the last bracket in format 1.
- V-15 (T) Delete the commas between data-name-1 through data-name-11, inclusively, in format 1.
- V-16 (T) Insert a terminal period following the last bracket in format 3.
- V-16 (T) Delete the commas between data-name-1 through data-name-6, inclusively, in format 3.
- V-27 (T) PERFORM format, AFTER phrase, change "literal-3" immediately after the reserved word AFTER to "index-name-3".
- V-28 (T) Insert "[END-REWRITE]" at the end of the first REWRITE statement.
- V-38 Change the format for subscripting to read:

$$\left\{ \begin{array}{l} \text{condition-name-1} \\ \text{data-name-1} \end{array} \right\} \left(\begin{array}{l} \text{integer-1} \\ \text{data-name-2} \left[\begin{array}{l} + \\ - \end{array} \right] \text{integer-2} \\ \text{index-name-1} \left[\begin{array}{l} + \\ - \end{array} \right] \text{integer-3} \end{array} \right) \dots$$

V-38 Add the following after the format for subscripting:

FUNCTION-IDENTIFIER:

FUNCTION function-name-1 [({argument-1} ...)] [reference-modifier]

V-38 Change the format for reference modification to read:

$\left\{ \begin{array}{l} \text{data-name-1} \\ \text{FUNCTION function-name-1 [({argument-1} ...)]} \end{array} \right\} (\text{leftmost-character-position:}(\text{length}))$

V-38

Change the format for identifier to read:

Format 1:

function-identifier-1

Format 2:

$$\text{data-name-1} \left[\begin{array}{c} \text{IN} \\ \text{OF} \end{array} \right] \text{data-name-2} \dots \left[\begin{array}{c} \text{IN} \\ \text{OF} \end{array} \right] \left\{ \begin{array}{l} \text{cd-name-1} \\ \text{file-name-1} \\ \text{report-name-1} \end{array} \right\}$$

[({subscript} ...)] [reference-modifier]

V-38

Insert after page V-38 the contents of pages A-16 and A-17 which contain the general format for functions.

GENERAL FORMAT FOR INTRINSIC FUNCTIONS

FUNCTION ACOS (argument-1)

FUNCTION ANNUITY (argument-1 argument-2)

FUNCTION ASIN (argument-1)

FUNCTION ATAN (argument-1)

FUNCTION CHAR (argument-1)

FUNCTION COS (argument-1)

FUNCTION CURRENT-DATE

FUNCTION DATE-OF-INTEGER (argument-1)

FUNCTION DAY-OF-INTEGER (argument-1)

FUNCTION FACTORIAL (argument-1)

FUNCTION INTEGER (argument-1)

FUNCTION INTEGER-OF-DATE (argument-1)

FUNCTION INTEGER-OF-DAY (argument-1)

FUNCTION INTEGER-PART (argument-1)

FUNCTION LENGTH (argument-1)

FUNCTION LOG (argument-1)

FUNCTION LOG10 (argument-1)

FUNCTION LOWER-CASE (argument-1)

FUNCTION MAX ({argument-1} ...)

FUNCTION MEAN ({argument-1} ...)

FUNCTION MEDIAN ({argument-1} ...)

FUNCTION MIDRANGE ({argument-1} ...)

FUNCTION MIN ({argument-1} ...)

FUNCTION MOD (argument-1 argument-2)

FUNCTION NUMVAL (argument-1)

FUNCTION NUMVAL-C (argument-1 [argument-2])

GENERAL FORMAT FOR INTRINSIC FUNCTIONS

FUNCTION ORD (argument-1)

FUNCTION ORD-MAX ({argument-1} ...)

FUNCTION ORD-MIN ({argument-1} ...)

FUNCTION PRESENT-VALUE (argument-1 {argument-2} ...)

FUNCTION RANDOM [(argument-1)]

FUNCTION RANGE ({argument-1} ...)

FUNCTION REM (argument-1 argument-2)

FUNCTION REVERSE (argument-1)

FUNCTION SIN (argument-1)

FUNCTION SORT (argument-1)

FUNCTION STANDARD-DEVIATION ({argument-1} ...)

FUNCTION SUM ({argument-1} ...)

FUNCTION TAN (argument-1)

FUNCTION UPPER-CASE (argument-1)

FUNCTION VARIANCE ({argument-1} ...)

FUNCTION WHEN-COMPILED

Page No.	Change to ANSI X3.23-1985 and ISO 1989-1985
VI-1	Paragraph 1.3, change title to read: "RESTRICTIONS ON OVERALL LANGUAGE".
VI-2	Renumber paragraphs 1.3.5 and 1.3.6 to 1.3.6 and 1.3.7, respectively; and add new paragraph 1.3.5 without boxing as follows: 1.3.5 Function-Identifier The availability of function-identifiers is dependent on whether the Intrinsic Function module is supported by the implementation.
VI-13 (T)	Paragraph 4.5.2, SYMBOLIC CHARACTERS clause, delete outermost set of braces.
VI-20 (T)	Paragraph 5.3.2 of the OCCURS clause, delete the box around the first occurrence of "[INDEXED BY {index-name-1} ...]".
VI-31 (T)	Paragraph 5.9.4 of the PICTURE clause, general rule 8, second line, change "explain" to "explained".
VI-50 (T)	Paragraph 5.15.6 of the VALUE clause, general rule 6, second line, change "or in a entry" to "or in an entry".
VI-57	Paragraph 6.3.1.2, replace the first paragraph after the general format beginning with "The usage of the operand..." by "Identifier-1 must reference a data item whose usage is explicitly or implicitly DISPLAY. If identifier-1 is a function-identifier, it must reference an alphanumeric function."
VI-62	Paragraph 6.3.4, penultimate paragraph: a. Insert "(2)" at the beginning of the paragraph and indent 3 spaces. b. Change the first sentence in part to read: "Values are established for arithmetic expressions and functions if and when..."
VI-70	Paragraph 6.4.7, change in part to read: "...PICTURE clause or function definition, then the result..."
VI-96	Paragraph 6.18.4, general rule 4, change in part to read: "If any identifier is subscripted or is a function-identifier, the subscript or function-identifier is evaluated..."
VI-103	Paragraph 6.19.4, general rule 2, second paragraph, change the first sentence to read: "If identifier-1 is reference modified, subscripted, or is a function-identifier, the reference modifier, subscript, or function-identifier is evaluated only once, immediately before data is moved to the first of the receiving operands."
VI-104 (T)	Paragraph 6.19.4 of the MOVE statement, general rule 3, fourth line, change "alphabetic, numeric edited," to "alphabetic, alphanumeric, numeric edited,"
VI-105 (T)	Paragraph 6.19.4 of the MOVE statement, general rule 4c, change indentation to align with general rule 4b.

- VII-6 (T) Paragraph 2.1, in the Input-Output Section of the Sequential I-O module, delete the box around the general format:
- [I-O-CONTROL. [input-output-control-entry]]
- VII-48 Paragraph 4.5.3, change syntax rule 1 to read:
- (1) If identifier-1 is a function-identifier, it must reference an alphanumeric function. When identifier-1 is not a function-identifier, record-name-1 and identifier-1 must not reference the same storage area.
- VII-52 Paragraph 4.7.3, change syntax rule 1 to read:
- (1) If identifier-1 is a function-identifier, it must reference an alphanumeric function. When identifier-1 is not a function-identifier, record-name-1 and identifier-1 must not reference the same storage area.
- VIII-30 Paragraph 4.6.3, change syntax rule 1 to read:
- (1) If identifier-1 is a function-identifier, it must reference an alphanumeric function. When identifier-1 is not a function-identifier, record-name-1 and identifier-1 must not reference the same storage area.
- VIII-30 (T) Paragraph 4.6.3, change syntax rule 4 in part to read: "...mode for which..."
- VIII-37 Paragraph 4.9.3, change syntax rule 1 to read:
- (1) If identifier-1 is a function-identifier, it must reference an alphanumeric function. When identifier-1 is not a function-identifier, record-name-1 and identifier-1 must not reference the same storage area.
- IX-3 (T) Paragraph 1.3.4, entitled "I-O Status", second occurrence of a paragraph numbered (1), subparagraphs c and d, change indentation to align with subparagraph b.
- IX-4 (T) Paragraph 1.3.4, entitled "I-O Status", second occurrence of a paragraph numbered (3), subparagraph b, first line, box "or rewrite".
- IX-6 (T) Paragraph 1.3.5, entitled "The Invalid Key Condition", second occurrence of a paragraph numbered 2, first line, change "If not exception" to "If no exception".
- IX-7 (T) Paragraph 1.3.7, first paragraph, last line, change "an" to "and".
- IX-33 Paragraph 4.6.3, change syntax rule 1 to read:
- (1) If identifier-1 is a function-identifier, it must reference an alphanumeric function. When identifier-1 is not a function-identifier, record-name-1 and identifier-1 must not reference the same storage area.
- IX-33 (T) Paragraph 4.6.3 of the REWRITE statement in the Indexed I-O module, syntax rule 3, change to read: "The INVALID KEY phrase must be specified in the REWRITE statement for indexed files for which an appropriate USE AFTER STANDARD EXCEPTION procedure is not specified."