INTERNATIONAL STANDARD

ISO/IEC 1989

Third edition
2023-01

Information technology — Programming languages, their environments and system software interfaces — Programming language COBOL

Technologies de l'information — Langages de programmation, leur environnement et interfaces des logiciels de systèmes — Langage de programmation COBOL



Reference number
ISO/IEC 1989:2023(E)

© ISO/IEC 2023



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2023

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

© ISO/IEC 2023 – All rights reserved

Contents

Contents iii

Tables xxiv

Figures xxv

Foreword xxvi

Introduction xxviii

1 Scope 1

2 Normative references 2

3 Terms and definitions 3

4 Conformance to this Working Draft International Standard 21
4.1 General 21
4.2 A conforming implementation 21
4.2.1 General 21
4.2.2 Acceptance of standard language elements 21
4.2.3 Interaction with non-COBOL runtime modules 21
4.2.4 Interaction between COBOL implementations 21
4.2.5 Implementor-defined language elements 22
4.2.6 Processor-dependent language elements 22
4.2.7 Optional language elements 22
4.2.8 Reserved words 23
4.2.9 Standard extensions 23
4.2.10 Nonstandard extensions 23
4.2.11 Substitute or additional language elements 24
4.2.12 Archaic language elements 24
4.2.13 Obsolete language elements 24
4.2.14 Externally-provided functionality 24
4.2.15 Limits 24
4.2.16 User documentation 25
4.2.17 Character substitution 25
4.3 A conforming compilation group 25
4.4 A conforming run unit 25
4.5 Relationship of a conforming compilation group to a conforming implementation 26
4.6 Relationship of a conforming run unit to a conforming implementation 26

5 Description techniques 27
5.1 General 27
5.2 General formats 27

5.2.1 General 27
5.2.2 Keywords 27
5.2.3 Optional words 28
5.2.4 Operands 28
5.2.5 Level numbers 28
5.2.6 Options 29
5.2.7 Ellipses 29
5.2.8 Punctuation 29
5.2.9 Special characters 30
5.2.10 Meta-terms 30
5.3 Rules 30
5.3.1 General 30
5.3.2 Syntax rules 30
5.3.3 General rules 30
5.3.4 Argument rules 30
5.3.5 Returned value rules 30
5.4 Arithmetic expressions 31
5.4.1 General 31
5.4.2 Textually subscripted operands 31
5.4.3 Ellipses 31
5.5 Integer operands 31
5.6 Informal description 32
5.7 Hyphens in text 32

6 Reference format 33
6.1 General 33
6.2 Indicators 33
6.2.1 General 33
6.2.2 Fixed indicators 34
6.2.3 Floating indicators 34
6.3 Fixed-form reference format 36
6.3.1 General 36
6.3.2 Sequence number area 36
6.3.3 Indicator area 36
6.3.4 Program-text area 37
6.3.5 Continuation of lines 37
6.3.6 Blank lines 38
6.3.7 Comments 38
6.4 Free-form reference format 38
6.4.1 General 38
6.4.2 Continuation of lines 39
6.4.3 Blank lines 39
6.4.4 Comments 39
6.5 Logical conversion 40

7 Compiler directing facility 42

7.1 General 42
7.2 Text manipulation 43
7.2.1 General 43
7.2.2 Text manipulation elements 44
7.2.3 COPY statement 46
7.2.4 REPLACE statement 50
7.3 Compiler directives 54
7.3.1 General 54
7.3.2 General format 54
7.3.3 Syntax rules 54
7.3.4 General rules 55
7.3.5 Conditional compilation 55
7.3.6 Compile-time arithmetic expressions 55
7.3.7 Compile-time boolean expressions 56
7.3.8 Constant conditional expression 57
7.3.9 CALL-CONVENTION directive 59
7.3.10 COBOL-WORDS directive 60
7.3.11 DEFINE directive 62
7.3.12 DISPLAY directive 64
7.3.13 EVALUATE directive 66
7.3.14 FLAG-02 directive 70
7.3.15 FLAG-14 directive 72
7.3.16 IF directive 75
7.3.17 LEAP-SECOND directive 76
7.3.18 LISTING directive 78
7.3.19 PAGE directive 79
7.3.20 POP directive 80
7.3.21 PROPAGATE directive 81
7.3.22 PUSH directive 82
7.3.23 REF-MOD-ZERO-LENGTH directive 83
7.3.24 SOURCE FORMAT directive 84
7.3.25 TURN directive 85

8 Language fundamentals 87
8.1 Character sets 87
8.1.1 General 87
8.1.2 Computer's coded character set 87
8.1.3 COBOL character repertoire 90
8.1.4 Alphabets 93
8.1.5 Collating sequences 93
8.2 Locales 94
8.2.1 General 94
8.2.2 Locale field names 95
8.3 Lexical elements 97
8.3.1 General 97
8.3.2 COBOL words 97

8.3.2.1 General 97
8.3.2.2 User-defined words 97
8.3.2.3 System-names 103
8.3.2.3.1 General 103
8.3.2.4 Reserved words 105
8.3.2.4.1 General 105
8.3.2.5 Context-sensitive words 106
8.3.2.6 Intrinsic-function-names 106
8.3.2.7 Exception-names 106
8.3.3 Literals 106
8.3.3.1 General 106
8.3.3.2 Alphanumeric literals 107
8.3.3.2.1 General 107
8.3.3.2.2 General format 107
8.3.3.2.3 Syntax rules 107
8.3.3.2.4 General rules 108
8.3.3.3 Numeric literals 109
8.3.3.4 Boolean literals 110
8.3.3.5 National literals 111
8.3.3.6 Figurative constant values 113
8.3.4 Picture character-strings 117
8.3.5 Separators 117
8.4 References 119
8.4.1 General 119
8.4.2 Uniqueness of reference 119
8.4.2.1 General 119
8.4.2.2 Qualification 119
8.4.2.3 Subscripts 122
8.4.2.3.1 General 122
8.4.2.3.2 General format 122
8.4.2.3.3 Syntax rules 123
8.4.2.3.4 General rules 124
8.4.3 Identifiers 124
8.4.3.1 Identifier 124
8.4.3.2 Function-identifier 127
8.4.3.3 Reference-modification 131
8.4.3.4 Inline method invocation 133
8.4.3.5 Object-view 134
8.4.3.6 EXCEPTION-OBJECT 135
8.4.3.7 NULL object reference 136
8.4.3.8 SELF and SUPER 136
8.4.3.9 Object property 137
8.4.3.10 NULL address pointer and message tag content 139
8.4.3.11 Data-address-identifier 139
8.4.3.12 Function-address-identifier 140
8.4.3.13 Program-address-identifier 141

8.4.3.14 LINAGE-COUNTER 142
8.4.3.15 Report counters 143
8.4.4 Condition-name 144
8.4.5 Explicit and implicit data item references 145
8.4.6 Scope of names 146
8.4.6.1 General 146
8.4.6.2 Local and global names 147
8.4.6.3 Scope of program-names 149
8.4.6.4 Scope of object-class-names and interface-names 149
8.4.6.5 Scope of method-names 149
8.4.6.6 Scope of function-prototype-names 150
8.4.6.7 Scope of user-function-names 150
8.4.6.8 Scope of program-prototype-names 150
8.4.6.9 Scope of compilation-variable-names 150
8.4.6.10 Scope of parameter-names 150
8.4.6.11 Scope of property-names 150
8.5 Data description and representation 151
8.5.1 Computer independent data description 151
8.5.1.1 General 151
8.5.1.2 Files and records 151
8.5.1.3 Levels 151
8.5.1.3.1 General 151
8.5.1.3.2 Level-numbers 152
8.5.1.3.3 Tables 152
8.5.1.4 Limitations of character handling 153
8.5.1.5 Algebraic signs 153
8.5.1.6 Alignment of data items in storage 154
8.5.1.6.3 Alignment of data items of usage bit 154
8.5.1.6.4 Item alignment for increased object-code efficiency 155
8.5.1.6.5 Alignment of strongly-typed group items 155
8.5.1.7 Fixed-capacity tables 156
8.5.1.9 Dynamic-capacity tables 156
8.5.1.10 Dynamic-length elementary items 158
8.5.1.10.1 General 158
8.5.1.10.2 Structure of a dynamic-length elementary item 158
8.5.1.10.3 Location of dynamic-length elementary items 158
8.5.1.10.4 Operations on dynamic-length elementary items 159
8.5.1.11 Variable-length data items 159
8.5.1.11.2 Contiguity of data items 159
8.5.1.11.3 Availability and persistence of variable-length data items 159
8.5.1.12 Variable-length groups 160
8.5.1.12.1 General 160
8.5.1.12.2 Positional correspondence 161
8.5.1.12.3 Matching 161
8.5.2 Class and category of data items and literals 161
8.5.2.1 General 161

8.5.2.2 Alphabetic category 162
8.5.2.3 Alphanumeric category 162
8.5.2.4 Alphanumeric-edited category 163
8.5.2.5 Boolean category 163
8.5.2.6 Data-pointer category 163
8.5.2.7 Function-pointer category 163
8.5.2.8 Index category 163
8.5.2.9 Message-tag category 164
8.5.2.10 National category 164
8.5.2.11 National-edited category 164
8.5.2.12 Numeric category 164
8.5.2.13 Numeric-edited category 165
8.5.2.14 Object-reference category 165
8.5.2.15 Program-pointer category 165
8.5.3 Types 165
8.5.3.1 General 165
8.5.3.2 Weakly-typed items 166
8.5.3.3 Strongly-typed group items 167
8.5.4 Zero-length items 167
8.6 Scope and life cycle of data 167
8.6.1 General 167
8.6.2 Global names and local names 167
8.6.3 External and internal items 168
8.6.4 Automatic, initial, and static internal items 168
8.6.5 Based entries and based data items 170
8.6.6 Common, initial, and recursive attributes 170
8.6.7 Sharing data items 171
8.7 Operators 172
8.7.1 Arithmetic operators 172
8.7.2 Boolean operators 172
8.7.3 Concatenation operator 173
8.7.4 Invocation operator 173
8.7.5 Relational operators 173
8.7.6 Logical operators 174
8.8 Expressions 175
8.8.1 Arithmetic expressions 175
8.8.2 Boolean expressions 182
8.8.3 Concatenation expressions 185
8.8.4 Conditional expressions 186
8.8.4.1 General 186
8.8.4.2 Simple relation conditions 186
8.8.4.5 Simple condition-name condition (conditional variable) 197
8.8.4.6 Simple switch-status condition 197
8.8.4.7 Simple sign condition 198
8.8.4.8 Simple omitted argument condition 199
8.8.4.9 Complex conditions 200

8.8.4.10 Complex negated conditions ... 200
8.8.4.11 Complex Combined conditions ... 201
8.8.4.12 Abbreviated combined relation conditions ... 202
8.8.4.13 Order of evaluation of conditions ... 204
8.9 Reserved words ... 205
8.10 Context-sensitive words ... 209
8.11 Intrinsic function names ... 213
8.12 Compiler-directive words ... 215
8.13 External repository ... 216

9 I-O, objects, and user-defined functions ... 217
9.1 Files ... 217
9.1.1 Physical and logical files ... 217
9.1.2 Record area ... 217
9.1.3 File connector ... 218
9.1.4 Open mode ... 218
9.1.5 Sharing file connectors ... 219
9.1.6 Fixed file attributes ... 219
9.1.7 Organization ... 219
9.1.7.1 General ... 219
9.1.7.2 Sequential ... 219
9.1.7.3 Relative ... 220
9.1.7.4 Indexed ... 220
9.1.8 Access modes ... 220
9.1.8.1 General ... 220
9.1.8.2 Sequential access mode ... 221
9.1.8.3 Random access mode ... 221
9.1.8.4 Dynamic access mode ... 221
9.1.9 Reel and unit ... 221
9.1.10 Current volume pointer ... 222
9.1.11 File position indicator ... 222
9.1.12 Input-output exception processing ... 222
9.1.13 I-O status ... 223
9.1.13.1 General ... 223
9.1.13.2 Successful completion ... 224
9.1.13.3 Implementor-defined successful completion ... 225
9.1.13.4 At end condition with unsuccessful completion ... 225
9.1.13.5 Invalid key condition with unsuccessful completion ... 226
9.1.13.6 Permanent error condition with unsuccessful completion ... 226
9.1.13.7 Logic error condition with unsuccessful completion ... 227
9.1.13.8 Record operation conflict condition with unsuccessful completion ... 228
9.1.13.9 File sharing conflict condition with unsuccessful completion ... 229
9.1.13.10 Record with invalid content with unsuccessful completion ... 229
9.1.13.11 Implementor-defined condition with unsuccessful completion ... 229
9.1.14 Invalid key condition ... 230
9.1.15 Sharing mode ... 230

9.1.16 Record locking . . . 232
9.1.17 Logical unit of work . . . 232
9.1.18 Commit and Rollback . . . 233
9.1.19 Sort file . . . 234
9.1.20 Merge file . . . 234
9.1.21 Dynamic file assignment . . . 235
9.1.22 Report file . . . 235
9.2 Screens . . . 236
9.2.1 Terminal screen . . . 236
9.2.2 Function keys . . . 236
9.2.3 CRT status . . . 236
9.2.4 Cursor . . . 237
9.2.5 Cursor locator . . . 238
9.2.6 Current screen item . . . 238
9.2.7 Color number . . . 238
9.3 Objects . . . 240
9.3.1 Objects and classes . . . 240
9.3.2 Object references . . . 240
9.3.3 Predefined object references . . . 240
9.3.4 Methods . . . 240
9.3.5 Polymorphism . . . 240
9.3.5.1 General . . . 240
9.3.5.2 Class polymorphism . . . 241
9.3.5.3 Parametric polymorphism . . . 241
9.3.6 Method invocation . . . 242
9.3.7 Method prototypes . . . 246
9.3.8 Conformance and interfaces . . . 246
9.3.8.1 General . . . 246
9.3.8.2 Conformance for object orientation . . . 247
9.3.8.2.1 General . . . 247
9.3.8.2.2 Interfaces . . . 247
9.3.8.2.3 Conformance between interfaces . . . 247
9.3.8.2.4 Conformance for parameterized classes and parameterized interfaces . . . 250
9.3.9 Class inheritance . . . 250
9.3.10 Interface inheritance . . . 251
9.3.11 Interface implementation . . . 251
9.3.12 Parameterized classes . . . 251
9.3.13 Parameterized interfaces . . . 252
9.3.14 Object life cycle . . . 252
9.3.14.1 General . . . 252
9.3.14.2 Life cycle for factory objects . . . 252
9.3.14.3 Life cycle for instance objects . . . 252
9.4 User-defined functions . . . 252

10 Structured compilation group . . . 254
10.1 General . . . 254

10.2 Compilation units . . . 254
10.3 Source units . . . 254
10.4 Contained source units . . . 255
10.5 Source elements and runtime elements . . . 255
10.6 COBOL compilation group . . . 256
10.6.1 General format . . . 256
10.6.2 Syntax rules . . . 259
10.6.3 General rule . . . 260
10.7 End markers . . . 261
10.7.1 General . . . 261
10.7.2 General format . . . 261
10.7.3 Syntax rules . . . 261
10.7.4 General rule . . . 262

11 Identification division . . . 263
11.1 General . . . 263
11.2 Identification division structure . . . 263
11.3 CLASS-ID paragraph . . . 264
11.3.1 General . . . 264
11.3.2 General format . . . 264
11.3.3 Syntax rules . . . 264
11.3.4 General rules . . . 264
11.4 FACTORY paragraph . . . 266
11.4.1 General . . . 266
11.4.2 General format . . . 266
11.4.3 Syntax rules . . . 266
11.4.4 General rules . . . 266
11.5 FUNCTION-ID paragraph . . . 267
11.5.1 General . . . 267
11.5.2 General format . . . 267
11.5.3 Syntax rule . . . 267
11.5.4 General rules . . . 267
11.6 INTERFACE-ID paragraph . . . 268
11.6.1 General . . . 268
11.6.2 General format . . . 268
11.6.3 Syntax rules . . . 268
11.6.4 General rules . . . 268
11.7 METHOD-ID paragraph . . . 269
11.7.1 General . . . 269
11.7.2 General format . . . 269
11.7.3 Syntax rules . . . 269
11.7.4 General rules . . . 270
11.8 OBJECT paragraph . . . 271
11.8.1 General . . . 271
11.8.2 General format . . . 271
11.8.3 Syntax rules . . . 271

11.8.4 General rules 271
11.9 OPTIONS paragraph 272
11.9.1 General 272
11.9.2 General format 272
11.9.3 Syntax rule 272
11.9.4 General rule 272
11.9.5 ARITHMETIC clause 272
11.9.6 DEFAULT ROUNDED clause 273
11.9.7 ENTRY-CONVENTION clause 274
11.9.8 FLOAT-BINARY clause 275
11.9.9 FLOAT-DECIMAL clause 275
11.9.10 INITIALIZE clause 276
11.9.11 INTERMEDIATE ROUNDING clause 278
11.10 PROGRAM-ID paragraph 280
11.10.1 General 280
11.10.2 General format 280
11.10.3 Syntax rules 280
11.10.4 General rules 281

12 Environment division 282
12.1 General 282
12.2 Environment division structure 282
12.3 Configuration section 283
12.3.1 General 283
12.3.2 General format 283
12.3.3 Syntax rules 283
12.3.4 General rule 283
12.3.5 SOURCE-COMPUTER paragraph 284
12.3.6 OBJECT-COMPUTER paragraph 285
12.3.7 SPECIAL-NAMES paragraph 289
12.3.8 REPOSITORY paragraph 304
12.4 Input-output section 310
12.4.1 General 310
12.4.2 General format 310
12.4.3 Syntax rule 310
12.4.4 FILE-CONTROL paragraph 311
12.4.5 File control entry 311
12.4.5.4 ACCESS MODE clause 319
12.4.5.6 ALTERNATE RECORD KEY clause 320
12.4.5.7 COLLATING SEQUENCE clause 322
12.4.5.8 FILE STATUS clause 324
12.4.5.9 LOCK MODE clause 325
12.4.5.10 ORGANIZATION clause 327
12.4.5.11 RECORD DELIMITER clause 328
12.4.5.12 RECORD KEY clause 329
12.4.5.13 RELATIVE KEY clause 330

12.4.5.14 RESERVE clause . . . 331
12.4.5.15 SHARING clause . . . 332
12.4.6 I-O-CONTROL paragraph . . . 333

13 Data division . . . 338
13.1 General . . . 338
13.2 Data division structure . . . 339
13.2.1 General format . . . 339
13.3 Explicit and implicit attributes . . . 339
13.4 File section . . . 341
13.4.1 General . . . 341
13.4.2 General format . . . 341
13.4.3 Syntax rule . . . 341
13.4.4 General rule . . . 341
13.4.5 File description entry . . . 342
13.4.6 Sort-merge file description entry . . . 346
13.5 Working-storage section . . . 347
13.5.1 General . . . 347
13.5.2 General format . . . 347
13.5.3 Syntax rule . . . 347
13.5.4 General rules . . . 347
13.6 Local-storage section . . . 348
13.6.1 General . . . 348
13.6.2 General format . . . 348
13.6.3 Syntax rule . . . 348
13.6.4 General rules . . . 348
13.7 Linkage section . . . 349
13.7.1 General . . . 349
13.7.2 General format . . . 349
13.7.3 Syntax rules . . . 349
13.7.4 General rules . . . 350
13.8 Report section . . . 351
13.8.1 General . . . 351
13.8.2 General format . . . 351
13.8.3 Syntax rule . . . 351
13.8.4 Report description entry . . . 351
13.8.5 Report group description entry . . . 351
13.8.6 Report subdivisions . . . 352
13.8.6.2 Physical subdivisions of a report . . . 352
13.8.6.2.1 Pages . . . 352
13.8.6.2.2 Lines . . . 352
13.8.6.2.3 Report Items . . . 352
13.8.6.3 Logical Subdivisions of a Report . . . 352
13.9 Screen section . . . 354
13.9.1 General . . . 354
13.9.2 General format . . . 354

13.9.3 Syntax rule 354
13.9.4 General rule 354
13.10 Constant entry 355
13.10.1 General 355
13.10.2 General format 355
13.10.3 Syntax rules 355
13.10.4 General rules 356
13.11 Record description entry 357
13.11.1 General 357
13.12 Type declaration entry 357
13.13 77-level data description entry 357
13.14 Report description entry 358
13.14.1 General 358
13.14.2 General format 358
13.14.3 Syntax rules 358
13.14.4 General rule 358
13.15 Report group description entry 359
13.15.1 General 359
13.15.2 General format 359
13.15.3 Syntax rules 360
13.15.4 General rules 361
13.16 Data description entry 362
13.16.1 General 362
13.16.2 General formats 363
13.16.3 Syntax rules 365
13.16.4 General rules 367
13.17 Screen description entry 368
13.17.1 General 368
13.17.2 General formats 368
13.17.3 Syntax rules 370
13.17.4 General rules 371
13.18 Data division clauses 372
13.18.1 ALIGNED clause 372
13.18.2 ANY LENGTH clause 373
13.18.3 AUTO clause 374
13.18.4 BACKGROUND-COLOR clause 375
13.18.5 BASED clause 376
13.18.6 BELL clause 377
13.18.7 BLANK clause 378
13.18.8 BLANK WHEN ZERO clause 379
13.18.9 BLINK clause 380
13.18.10 BLOCK CONTAINS clause 381
13.18.11 CLASS clause 382
13.18.12 CODE clause 383
13.18.13 CODE-SET clause 384
13.18.14 COLUMN clause 386

13.18.15 CONSTANT RECORD clause ... 391
13.18.16 CONTROL clause ... 392
13.18.17 DEFAULT clause ... 394
13.18.18 DESTINATION clause ... 396
13.18.19 DYNAMIC LENGTH clause ... 397
13.18.20 Entry-name clause ... 398
13.18.21 ERASE clause ... 399
13.18.22 EXTERNAL clause ... 400
13.18.23 FOREGROUND-COLOR clause ... 402
13.18.24 FORMAT clause ... 403
13.18.25 FROM clause ... 406
13.18.26 FULL clause ... 407
13.18.27 GLOBAL clause ... 408
13.18.28 GROUP INDICATE clause ... 409
13.18.29 GROUP-USAGE clause ... 410
13.18.30 HIGHLIGHT clause ... 412
13.18.31 INVALID clause ... 413
13.18.32 JUSTIFIED clause ... 414
13.18.33 Level-number ... 415
13.18.34 LINAGE clause ... 417
13.18.35 LINE clause ... 420
13.18.36 LOWLIGHT clause ... 426
13.18.37 NEXT GROUP clause ... 427
13.18.38 OCCURS clause ... 430
13.18.39 PAGE clause ... 438
13.18.40 PICTURE clause ... 441
13.18.41 PRESENT WHEN clause ... 461
13.18.42 PROPERTY clause ... 464
13.18.43 RECORD clause ... 467
13.18.44 REDEFINES clause ... 471
13.18.45 RENAMES clause ... 473
13.18.46 REPORT clause ... 475
13.18.47 REQUIRED clause ... 476
13.18.48 REVERSE-VIDEO clause ... 477
13.18.49 SAME AS clause ... 478
13.18.50 SECURE clause ... 480
13.18.51 SELECT WHEN clause ... 481
13.18.52 SIGN clause ... 483
13.18.54 SUM clause ... 487
13.18.55 SYNCHRONIZED clause ... 491
13.18.56 TO clause ... 493
13.18.57 TYPE clause ... 494
13.18.58 TYPEDEF clause ... 500
13.18.59 UNDERLINE clause ... 501
13.18.60 USAGE clause ... 502
13.18.61 USING clause ... 512

13.18.62 VALIDATE-STATUS clause 513
13.18.63 VALUE clause 516
13.18.64 VARYING clause 525

14 Procedure division 527
14.1 General 527
14.2 Procedure division structure 527
14.2.1 General formats 527
14.2.2 Syntax rules 528
14.2.3 General rules 529
14.3 Declaratives 532
14.4 Procedures 532
14.4.1 General 532
14.4.2 Sections 532
14.4.3 Paragraphs 532
14.5 Procedural statements and sentences 532
14.5.1 General 532
14.5.2 Conditional phrase 535
14.5.3 Scope of statements 535
14.5.3.1 General 535
14.5.3.2 Explicit scope termination 535
14.5.3.3 Implicit scope termination 535
14.6 Execution 536
14.6.1 Run unit organization 536
14.6.2 State of a function, method, object, or program 537
14.6.2.1 General 537
14.6.2.2 Active state 537
14.6.2.3 Initial and last-used states of data 537
14.6.2.3.1 General 537
14.6.2.3.2 Initial state 538
14.6.2.3.3 Last-used state 538
14.6.2.4 Initial state of object data 539
14.6.3 Explicit and implicit transfers of control 539
14.6.4 Item identification 540
14.6.5 Results of runtime element execution 541
14.6.6 Locale identification 541
14.6.7 Sending and receiving operands 542
14.6.8 Alignment and transfer of data into data items 542
14.6.8.1 General 542
14.6.8.2 Fixed-point numeric and fixed-point numeric-edited receiving data items 542
14.6.8.3 Floating-point numeric receiving data items 543
14.6.8.4 Floating-point numeric-edited receiving data items 543
14.6.8.5 Receiving data items of categories alphabetic, alphanumeric, alphanumeric-edited, national, and national edited 543
14.6.8.6 Receiving data items of category boolean 543
14.6.9 Operations on dynamic-capacity tables 544

14.6.10 Overlapping operands 545
14.6.11 Normal run unit termination 545
14.6.12 Abnormal run unit termination 546
14.6.13 Exception condition handling 546
14.6.13.1 Exception conditions 546
14.6.13.1.1 General 546
14.6.13.1.2 Normal completion of a declarative procedure 548
14.6.13.1.3 Fatal exception conditions 549
14.6.13.1.4 Nonfatal exception conditions 550
14.6.13.1.5 Exception objects 550
14.6.13.1.6 Exception-names and exception conditions 551
14.6.13.2 Incompatible data 558
14.7 Common phrases and features for statements 559
14.7.1 General 559
14.7.2 At end condition 560
14.7.3 Invalid key condition 560
14.7.4 ROUNDED phrase 560
14.7.5 SIZE ERROR phrase and size error condition 561
14.7.6 CORRESPONDING phrase 563
14.7.7 Arithmetic statements 564
14.7.8 THROUGH phrase 566
14.7.9 RETRY phrase 567
14.8 Conformance for parameters, returning items and external items 568
14.8.1 General 568
14.8.2 Parameters 568
14.8.2.1 General 568
14.8.2.2 Group items 569
14.8.2.3 Elementary items 569
14.8.2.3.1 General 569
14.8.2.3.2 Elementary items passed by reference 569
14.8.2.3.3 Elementary items passed by content or by value 571
14.8.3 Returning items 572
14.8.3.1 General 572
14.8.3.2 Group items 572
14.8.3.3 Elementary items 573
14.8.4 External items 574
14.9 Statements 576
14.9.1 ACCEPT statement 576
14.9.2 ADD statement 583
14.9.3 ALLOCATE statement 586
14.9.4 CALL statement 588
14.9.5 CANCEL statement 595
14.9.6 CLOSE statement 597
14.9.7 COMMIT statement 601
14.9.8 COMPUTE statement 602
14.9.9 CONTINUE statement 604

14.9.10 DELETE statement 605
14.9.11 DISPLAY statement 610
14.9.12 DIVIDE statement 614
14.9.13 EVALUATE statement 618
14.9.14 EXIT statement 623
14.9.15 FREE statement 627
14.9.16 GENERATE statement 628
14.9.17 GO TO statement 630
14.9.18 GOBACK statement 631
14.9.19 IF statement 635
14.9.20 INITIALIZE statement 637
14.9.21 INITIATE statement 642
14.9.22 INSPECT statement 643
14.9.23 INVOKE statement 651
14.9.24 MERGE statement 657
14.9.25 MOVE statement 664
14.9.26 MULTIPLY statement 673
14.9.27 OPEN statement 675
14.9.28 PERFORM statement 682
14.9.29 RAISE statement 691
14.9.30 READ statement 692
14.9.31 RECEIVE statement 702
14.9.32 RELEASE statement 704
14.9.33 RESUME statement 706
14.9.34 RETURN statement 708
14.9.35 REWRITE statement 710
14.9.36 ROLLBACK statement 718
14.9.37 SEARCH statement 720
14.9.38 SEND statement 726
14.9.39 SET statement 729
14.9.40 SORT statement 745
14.9.41 START statement 754
14.9.42 STOP statement 758
14.9.43 STRING statement 759
14.9.44 SUBTRACT statement 762
14.9.45 SUPPRESS statement 765
14.9.46 TERMINATE statement 766
14.9.47 UNLOCK statement 768
14.9.48 UNSTRING statement 769
14.9.49 USE statement 774
14.9.50 VALIDATE statement 780
14.9.51 WRITE statement 785

15 Intrinsic functions 796
15.1 General 796
15.2 Types of functions 796

15.3 Arguments . . . 796
15.3.1 Format arguments to international date and time functions . . . 799
15.3.1.1 General . . . 799
15.3.1.2 Calendar date formats . . . 799
15.3.1.3 Permissible values for data associated with calendar date formats . . . 799
15.3.1.4 Ordinal date formats . . . 799
15.3.1.5 Permissible values for data associated with ordinal date formats . . . 800
15.3.1.6 Week date formats . . . 800
15.3.1.7 Permissible values for data associated with week date formats . . . 800
15.3.2 Time formats . . . 800
15.3.3 Common time formats . . . 801
15.3.3.1 Common time formats with integer seconds representation . . . 801
15.3.3.2 Common time formats with fractional seconds representation . . . 801
15.3.3.3 Permissible values for data associated with common time formats . . . 802
15.3.3.4 Local time formats . . . 802
15.3.3.5 UTC time formats . . . 802
15.3.3.6 Offset time formats . . . 802
15.3.3.7 Combined date and time formats . . . 803
15.4 Returned values . . . 804
15.4.1 Numeric and integer functions . . . 804
15.5 Date and time conversion functions . . . 805
15.5.1 General . . . 805
15.5.2 Integer date form . . . 805
15.5.3 Standard date form . . . 806
15.5.4 Julian date form . . . 806
15.5.5 Standard numeric time form . . . 806
15.6 Summary of functions . . . 806
15.7 ABS function . . . 816
15.8 ACOS function . . . 817
15.9 ANNUITY function . . . 818
15.10 ASIN function . . . 819
15.11 ATAN function . . . 820
15.12 BASECONVERT function . . . 821
15.12.2 General format . . . 821
15.13 BOOLEAN-OF-INTEGER function . . . 822
15.14 BYTE-LENGTH function . . . 823
15.15 CHAR function . . . 825
15.16 CHAR-NATIONAL function . . . 826
15.17 COMBINED-DATETIME function . . . 827
15.18 CONCAT function . . . 828
15.19 CONVERT function . . . 829
15.20 COS function . . . 833
15.21 CURRENT-DATE function . . . 834
15.22 DATE-OF-INTEGER function . . . 836
15.23 DATE-TO-YYYYMMDD function . . . 837
15.24 DAY-OF-INTEGER function . . . 839

15.25 DAY-TO-YYYYDDD function 840
15.26 DISPLAY-OF function 842
15.27 E function 843
15.28 EXCEPTION-FILE function 844
15.29 EXCEPTION-FILE-N function 846
15.30 EXCEPTION-LOCATION function 848
15.31 EXCEPTION-LOCATION-N function 850
15.32 EXCEPTION-STATEMENT function 852
15.33 EXCEPTION-STATUS function 853
15.34 EXP function 854
15.35 EXP10 function 855
15.36 FACTORIAL function 856
15.37 FIND-STRING function 857
15.38 FORMATTED-CURRENT-DATE function 858
15.39 FORMATTED-DATE function 859
15.40 FORMATTED-DATETIME function 860
15.41 FORMATTED-TIME function 862
15.42 FRACTION-PART function 864
15.43 HIGHEST-ALGEBRAIC function 865
15.44 INTEGER function 867
15.45 INTEGER-OF-BOOLEAN function 868
15.46 INTEGER-OF-DATE function 869
15.47 INTEGER-OF-DAY function 870
15.48 INTEGER-OF-FORMATTED-DATE function 871
15.49 INTEGER-PART function 872
15.50 LENGTH function 873
15.51 LOCALE-COMPARE function 875
15.52 LOCALE-DATE function 876
15.53 LOCALE-TIME function 877
15.54 LOCALE-TIME-FROM-SECONDS function 878
15.55 LOG function 879
15.56 LOG10 function 880
15.57 LOWER-CASE function 881
15.58 LOWEST-ALGEBRAIC function 882
15.59 MAX function 884
15.60 MEAN function 885
15.61 MEDIAN function 886
15.62 MIDRANGE function 887
15.63 MIN function 888
15.64 MOD function 889
15.65 MODULE-NAME function 890
15.66 NATIONAL-OF function 892
15.67 NUMVAL function 893
15.68 NUMVAL-C function 895
15.69 NUMVAL-F function 898
15.70 ORD function 900

15.71 ORD-MAX function . . . 901
15.72 ORD-MIN function . . . 902
15.73 PI function . . . 903
15.74 PRESENT-VALUE function . . . 904
15.75 RANDOM function . . . 905
15.75.2 General format . . . 905
15.76 RANGE function . . . 906
15.77 REM function . . . 907
15.78 REVERSE function . . . 908
15.79 SECONDS-FROM-FORMATTED-TIME function . . . 909
15.80 SECONDS-PAST-MIDNIGHT function . . . 910
15.81 SIGN function . . . 911
15.82 SIN function . . . 912
15.83 SMALLEST-ALGEBRAIC function . . . 913
15.84 SQRT function . . . 915
15.85 STANDARD-COMPARE function . . . 916
15.86 STANDARD-DEVIATION function . . . 918
15.87 SUBSTITUTE function . . . 919
15.88 SUM function . . . 921
15.89 TAN function . . . 922
15.90 TEST-DATE-YYYYMMDD function . . . 923
15.91 TEST-DAY-YYYYDDD function . . . 924
15.92 TEST-FORMATTED-DATETIME function . . . 925
15.93 TEST-NUMVAL function . . . 926
15.94 TEST-NUMVAL-C function . . . 928
15.95 TEST-NUMVAL-F function . . . 930
15.96 TRIM function . . . 932
15.97 UPPER-CASE function . . . 934
15.98 VARIANCE function . . . 936
15.99 WHEN-COMPILED function . . . 937
15.100 YEAR-TO-YYYY function . . . 939

16 Standard classes . . . 941
16.1 General . . . 941
16.2 BASE class . . . 941
16.2.1 New method . . . 941
16.2.2 FactoryObject method . . . 942

A Language element lists . . . 943
A.1 Implementor-defined language element list . . . 943
A.2 Undefined language element list . . . 963
A.3 Processor-dependent language element list . . . 969
A.4 Optional language element list . . . 974

B Characters permitted in user-defined words . . . 980
B.1 General . . . 980

B.2 Notation 980
B.3 Repertoire of characters permitted in user-defined words 980

C Mapping of uppercase letters to lowercase letters in the COBOL character repertoire 998
C.1 Notations 998
C.2 General case mappings 998

D Concepts 1005
D.1 General 1005
D.2 Files 1005
D.3 Tables and dynamic-length elementary items 1027
D.4 Shared memory area 1038
D.5 Sharing of storage among data items 1038
D.6 Compilation group and run unit organization and communication 1041
D.7 Intrinsic function facility 1059
D.8 Types 1061
D.9 Addresses and pointers 1065
D.10 Boolean support and bit manipulation 1067
D.11 Character sets 1072
D.12 COBOL-WORDS directive 1075
D.13 Collating sequences 1077
D.14 Culturally-specific, culturally-adaptable, and multilingual applications 1082
D.15 External switches 1090
D.16 Common exception processing 1091
D.17 Rounding 1095
D.18 Forms of arithmetic 1099
D.19 Object oriented concepts 1104
D.20 Report writer 1130
D.21 Structured constant 1140
D.22 Validate facility 1141
D.23 Conditional expressions 1146
D.24 Examples of the use of the EDITING phrase 1150
D.25 Examples of the execution of the INSPECT statement 1151
D.26 Examples of the execution of the PERFORM statement with the VARYING phrase specified . 1155
D.27 Example of free-form reference format 1159
D.28 Conditional compilation 1160
D.29 CALL-CONVENTION directive 1162
D.30 ENTRY-CONVENTION clause 1162
D.31 Date and time handling 1162
D.32 Alternatives to HIGHEST-ALGEBRAIC, LOWEST-ALGEBRAIC and SMALLEST-ALGEBRAIC FUNCTIONS 1169

E Substantive changes list 1172
E.1 General 1172
E.2 Substantive changes potentially affecting existing programs 1172

E.3 Substantive changes probably not affecting existing programs 1181

F Archaic and obsolete language element lists 1199
F.1 Archaic language elements 1199
F.2 Obsolete language elements 1199

G Known errors 1201
G.1 Rationale 1201
G.2 List of errors 1201

BIBLIOGRAPHY 1203

Index 1204

Tables

1 COBOL character repertoire 90
2 Class and category relationships for elementary data items 162
3 Combinations of symbols in arithmetic expressions 175
4 Combination of symbols in boolean expressions 183
5 Combinations of conditions, logical operators, and parentheses 202
6 Relationship of alphabet-name to coded character set and collating sequence 297
7 Category and type of editing 453
8 Results of fixed insertion editing 454
9 Results of floating insertion editing 455
10 Format 1 picture symbol order of precedence 459
11 Format 2 picture symbol order of precedence 460
12 Procedural statements 533
13 Exception-names and exception conditions 552
14 Relationship of categories of physical files and the format of the CLOSE statement 598
15 Combination of operands in the EVALUATE statement 620
16 Validity of types of MOVE statements 666
17 Category of figurative constants used in the MOVE statement 671
18 Opening available and unavailable files (file not currently open) 676
19 Opening available shared files that are currently open by another file connector 677
20 Permissible I-O statements by access mode and open mode 679
21 Table of functions 807
A.1 Summary of record lock acquisition and release 1017
A.2 Examples of boolean operations 1069
A.3 ROUNDED MODE examples 1096

Figures

1 Fixed-form reference format 36
D.1 Format 1 SEARCH statement having two WHEN phrases 1033
D.2 Compilation group sample structure example 1042
D.3 Compilation group and run unit structures 1044
D.4 Manager class 1112
D.5 Banking hierarchy 1113
D.6 Example of page layout 1131
D.7 Evaluation of the condition-1 AND condition-2 AND ... condition-n 1146
D.8 Evaluation of the condition-1 OR condition-2 OR ... condition-n 1146
D.9 Evaluation of condition-1 OR condition-2 AND condition-3 1147
D.10 Evaluation of (condition-1 OR NOT condition-2) AND condition-3 AND condition-4 1148
D.11 The VARYING phrase of a PERFORM statement with the TEST BEFORE phrase having one condition 1155
D.12 The VARYING phrase of a PERFORM statement with the TEST BEFORE phrase having two conditions 1155
D.13 The VARYING phrase of a PERFORM statement with the TEST AFTER phrase having one condition 1156
D.14 The VARYING phrase of a PERFORM statement with the TEST AFTER phrase having two conditions 1157

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

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 copyrights or patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see https://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

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

This third edition cancels and replaces the second edition (ISO/IEC 1989:2014), which has been technically revised.

The main changes are as follows:

- The following were general enhancements:
  - An asynchronous messaging facility using the SEND statement and RECEIVE statement
  - Boolean exclusive or operators
  - Boolean shifting operators
  - COBOL words may now be 63 characters long
  - The PERFORM statement has been enhanced to specify a time period for pausing the program
  - A DELETE FILE statement
  - A nonfatal EC-I-O-WARNING exception condition to handle warnings for successful input-output statements
  - EXTERNAL attributes checking between programs
  - Infinite loop for the PERFORM statement using the UNTIL EXIT phrase

  - Inline exception handling using the exception-checking format of the PERFORM statement
  - An Enhanced INSPECT statement to inspect backwards
  - Line Sequential file organization
  - The SET statement has been enhanced to allow the setting of the length of a dynamic length elementary item
  - Alternate key suppression on indexed files using the SUPPRESS WHEN phrase of the ALTERNATE RECORD KEY clause
  - An optional Commit and rollback processing facility using the COMMIT statement and ROLLBACK statement
  - Unsigned Packed-Decimal items defined by the NO SIGN phrase of the USAGE clause
  - User-defined PICTURE clause editing using the EDITING phrase of the PICTURE clause
  - VALUE clause enhancements and changes for numeric-edited items
  - Type declarations may now be external items

- The following intrinsic functions were added or enhanced:
  - BASECONVERT function
  - CONCAT function
  - CONVERT function
  - EXCEPTION-FILE function and EXCEPTION-FILE-N function
  - FIND-STRING function
  - MODULE-NAME function
  - SMALLEST-ALGEBRAIC function
  - SUBSTITUTE function
  - TRIM function

- Additional compiler directives were added:
  - COBOL-WORDS directive
  - DISPLAY directive
  - FLAG-14 directive
  - POP directive
  - PUSH directive
  - REF-MOD-ZERO-LENGTH directive

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

COBOL began as a business programming language, but its present use has spread well beyond that to a general purpose programming language.

Any organization interested in reproducing the COBOL standard and specifications in whole or in part, using ideas from this document as the basis for an instruction manual or for any other purpose, is free to do so. However, all such organizations are requested to reproduce the following acknowledgment paragraphs in their entirety as part of the preface to any such publication (any organization using a short passage from this document, such as in a book review, is requested to mention "COBOL" in acknowledgment of the source, but need not quote the acknowledgment):

COBOL is an industry language and is not the property of any company or group of companies, or of any organization or group of organizations.

No warranty, expressed or implied, is made by any contributor or by the CODASYL COBOL Committee as to the accuracy and functioning of the programming system and language. Moreover, no responsibility is assumed by any contributor, or by the committee, in connection therewith.

The authors and copyright holders of the copyrighted materials used herein:

— FLOW-MATIC™[1];

— IBM®[2] Commercial Translator Form No F 28-8013, copyrighted 1959 by IBM;

— FACT©, DSI 27A5260-2760, copyrighted 1960 by Minneapolis-Honeywell

have specifically authorized the use of this material in whole or in part, in the COBOL specifications. Such authorization extends to the reproduction and use of COBOL specifications in programming manuals or similar publications.

For more details and additional changes, see E.2, Substantive changes potentially affecting existing programs and E.3, Substantive changes probably not affecting existing programs.

Further development of the COBOL language is a continuing process to provide facilities to satisfy user demand for the improved usability of the language and the adoption of relevant advances in techniques developed in the computer industry as a whole, including the desirability of interoperability with a wide variety of operating systems and other programming languages to enable developers to take advantage of their facilities, including pre-existing task solutions that then don't need to be repeated.

Annexes A, Language element lists B, Characters permitted in user-defined words, and C, Mapping of uppercase letters to lowercase letters in the COBOL character repertoire form a normative part of this document. Annexes D through G are for information only.

[1] FLOW-MATIC™ **i**s the trademark of a product supplied by Sperry Rand Corporation. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC of the product named.
[2] IBM is the trademark of International Business Machines Corporation. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC of the product named.

Annex D, Concepts, includes an explanation of major features as well as the more complicated prior features and is the suggested starting point for the reading of this document.

A complete list of technical changes is given in Annex E, Substantive changes list.

The previous COBOL standard was published in 2014. Implementors have provided language extensions in response to the demands of their users. Several changes and extensions have, therefore, been made in this document to prevent further divergence, and to ensure consistency among, and coherence within, various implementations.

Development of the COBOL language began before the invention of formal techniques for specification of programming languages. Hence, the COBOL standard uses its own description techniques, which are described in Clause 5, Description techniques. These techniques involve general formats, which describe the syntax, and natural language.

During the development of this document, great care was taken to minimize changes that would affect existing programs. Most substantive changes that potentially affect existing programs were introduced to resolve ambiguities in the previous COBOL standard. Details of the substantive changes are given in Annex E, Substantive changes list.

In this document, the following verbal forms are used:

— ‘shall’ indicates a requirement;
— ‘should’ indicates a recommendation;
— ‘can’ indicates a possibility or a capability;
— ‘may’ indicates a permission.

Information marked as ‘NOTE’ is intended to assist the understanding or use of the document. ‘Notes to entry’ used in Clause 3 provide additional information that supplements the terminological data and can contain requirements relating to the use of a term.

Information technology — Programming languages, their environments and system software interfaces — Programming language COBOL

1 Scope

This document specifies the syntax and semantics of COBOL. Its purpose is to promote a high degree of machine independence to permit the use of COBOL on a variety of data processing systems.

This document specifies:

— The form of a compilation group written in COBOL.

— The effect of compiling a compilation group.

— The effect of executing run units.

— The elements of the language for which a conforming implementation is required to supply a definition.

— The elements of the language for which meaning is explicitly undefined.

— The elements of the language that are dependent on the capabilities of the processor.

This document does not specify:

— The means whereby a compilation group written in COBOL is compiled into code executable by a processor.

— The time at which method, function, or program runtime modules are linked or bound to an activating statement, except that runtime binding occurs of necessity when the identification of the appropriate program or method is not known at compile time.

— The time at which parameterized classes and interfaces are expanded.

— The mechanism by which locales are defined and made available on a processor.

— The form or content of error, flagging, or warning messages.

— The form and content of listings produced during compilation, if any.

— The form of documentation produced by an implementor of products conforming to this document.

— The sharing of objects and resources other than files among run units.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 60559:2020, *Information technology — Microprocessor systems — Floating-Point Arithmetic*

ISO/IEC 646, *Information technology — ISO 7-bit coded character set for information interchange*

ISO/IEC 1001:2012, *Information technology — File structure and labelling of magnetic tapes for information interchange*

ISO 8601-1:2019, *Date and time — Representations for information interchange — Part 1: Basic rules*

ISO/IEC/IEEE 9945:2009, *Information technology — Portable Operating System Interface (POSIX®) Base Specifications, Issue 7*

ISO/IEC 10646, *Information technology — Universal Coded Character Set (UCS)*

ISO/IEC 14651:2020, *Information technology — International string ordering and comparison — Method for comparing character strings and description of the common template tailorable ordering*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at https://www.iso.org/obp
- IEC Electropedia: available at https://www.electropedia.org/

3.1
absolute item
item in a report that has a fixed position on a page

3.2
activated runtime element
function, method, or program placed into the active state

3.3
activating statement
statement that causes the execution of a function, method, or program

3.4
activating runtime element
function, method, or program that executed a given activating statement

3.5
active state
state of a function, method, or program that has been activated but has not yet returned control to the activating runtime element

3.6
alphabetic character
basic letter or a space character in the COBOL character repertoire

3.7
alphanumeric character
coded character in an alphanumeric coded character set, whether or not there is an assigned graphic symbol for that coded character

3.8
alphanumeric character position
<storage required> amount of physical storage required to store, or presentation space required to print or display, a single character of an alphanumeric character set

3.9
alphanumeric character position
<location within an item> location within an alphanumeric data item of an alphanumeric character

3.10
alphanumeric character set
alphanumeric coded character set
coded character set that the implementor has designated for representation of data items of usage display and alphanumeric literals

3.11
alphanumeric group item
group item except for a bit group item, a national group item, a strongly-typed group item, or a variable-length group item

3.12
argument
operand specified in an activating statement that specifies the data to be passed

3.13
assumed decimal point
decimal point position that does not involve the existence of an actual character in a data item and has logical meaning with no physical representation

3.14
based data item
data item established by association of a based entry with an actual data item or allocated storage

3.15
based entry
data description entry that serves as a template that is dynamically associated with data items or allocated storage

3.16
basic letter
uppercase letter 'A' through 'Z' or lowercase letter 'a' through 'z' in the COBOL character repertoire

3.17
big-endian
characterized by the arrangement of data within a data item such that its most significant component occupies the lowest (leftmost) component memory address within the item

3.18
bit
smallest unit in a computer's storage structure capable of representing two distinct alternatives

3.19
bit data item
elementary data item of category boolean and usage bit or a bit group item

3.20
block
physical record
physical unit of data that is normally composed of one or more logical records

3.21
boolean character
unit of information that consists of the value zero or one

3.22
boolean data item
data item capable of containing a boolean value

3.23
boolean expression
expression consisting of one or more boolean operands separated by boolean operator

3.24
boolean position
<storage required> amount of physical storage required to store, or presentation space required to print or display, a single boolean character

3.25
boolean position
<location within an item> the location within a boolean data item of a boolean character

3.26
boolean value
value consisting of a sequence of one or more boolean characters

3.27
byte
sequence of bits representing the smallest addressable character unit in the memory of a given computer

3.28
character boundary
leftmost bit of an addressing boundary in the storage of the computer

3.29
character position
<storage required> amount of physical storage required to store, or presentation space required to print or display, a single character – either an alphanumeric character or a national character

Note 1 to entry: As an example, each element of a combining sequence in the UCS occupies one character position. A UTF-16 surrogate pair occupies two character positions.

3.30
character position
<location within an item> location within an alphanumeric or national data item of a corresponding alphanumeric or national character

3.31
character-string
sequence of contiguous characters that form a COBOL word, a literal, or a picture character-string

3.32
class
<in object orientation> entity that defines common behavior and implementation for zero, one, or more objects

3.33
class
<of a data item> set of data items having common attributes or a common range of values, defined by the specific clauses in a data description entry; by the definition of a predefined identifier; or by the definition of an intrinsic function

Note 1 to entry: These common attributes or a common range of values are defined by the PICTURE clause (see 13.18.40, PICTURE clause), the USAGE clause, (see 13.18.60, USAGE clause) or the PICTURE and USAGE clauses.

3.34
class
<of data values> set of data values that are permissible in the content of a data item

3.35
class definition
compilation unit that defines a class of objects

3.36
clause
ordered set of consecutive COBOL character-strings whose purpose is to specify an attribute of an entry

3.37
COBOL character repertoire
repertoire of characters used in writing the syntax of a COBOL compilation group, except for comments and the content of non-hexadecimal alphanumeric and national literals

3.38
combining character
member of the Universal Coded Character Set that is intended for combination with the preceding non-combining graphic character or with a sequence of combining characters preceded by a non-combining character

[SOURCE: ISO/IEC 10646:2020, 3.44, modified — wording changes to COBOL usage]

3.39
common program
program that, despite being directly contained within another program, can be called from any program directly or indirectly contained in that other program

3.40
compilation group
sequence of one or more compilation units submitted together for compilation

3.41
compilation unit
source unit that is not nested within another source unit

3.42
composite sequence
sequence of graphic characters consisting of a base character followed by one or more combining characters

[SOURCE: ISO/IEC 10646:2020, 3.16, modified — wording changes to COBOL usage]

3.43
conditional statement
statement for which the truth value of a specified condition is evaluated and used to determine subsequent flow of control

3.44
conformance
<for object orientation> unidirectional relation that allows an object to be used according to an interface other than the interface of its own class

3.45
conformance
<for parameters> requirements for the relationship between arguments and formal parameters and between returning items in activating and activated runtime elements

3.46
control function
action that affects the recording, processing, transmission, or interpretation of data, and that has a coded representation consisting of one or more bytes

[SOURCE: ISO/IEC 10646:2020, 3.18, modified — wording changes to COBOL usage]

3.47
cultural element
element of data for computer use that can vary dependent on language, geographical territory, or other cultural circumstances

3.48
currency sign
COBOL character '$', used as the default currency symbol in a picture character-string and as the default currency string that appears in the edited format of data items

Note 1 to entry: Features exist for selection of other currency strings and currency symbols.

3.49
currency string
set of characters to be placed into numeric-edited data items as a result of editing operations when the item includes a currency symbol in its picture character-string

3.50
currency symbol
character used in a picture character-string to represent the presence of a currency string

3.51
current record
record that is available in the record area associated with a file

3.52
current volume pointer
conceptual entity that points to the current volume of a sequential file

3.53
data item
unit of data defined by a data description entry or resulting from the evaluation of an identifier

3.54
decimal point
decimal separator
character used to represent the radix point

3.55
declarative statement
USE statement that defines the conditions under which the procedures that follow the statement will be executed

3.56
de-editing
logical removal of all editing characters from a numeric-edited data item in order to determine that item's numeric value

3.57
delete file statement
FILE format of the DELETE statement

3.58
delete record statement
RECORD format of the DELETE statement

3.59
delimited scope statement
statement that is terminated by its explicit scope terminator

3.60
digit position
<storage required> amount of physical storage required to store, or presentation space required to print or display, a single digit

3.61
digit position
<location within an item> location within a numeric data item of a digit

3.62
dynamic access
access mode in which specific logical records can be obtained from or placed into a mass storage file in a nonsequential manner and obtained from a file in a sequential manner

3.63
dynamic storage
storage that is allocated and released on request during runtime

3.64
end marker
marker for the end of a source unit

3.65
endianness
ordering of individually addressable components of a given size within a data item of a larger size

3.66
entry
descriptive set of consecutive clauses terminated by a separator period

3.67
entry convention
information required to interact successfully with a function, method, or program

3.68
exception condition
condition detected at runtime that indicates that an error or exception to normal processing has occurred

3.69
exception object
object that acts as an exception condition

3.70
exception processing procedures
procedures within WHEN phrases within an exception-checking PERFORM statement or within a USE declarative

3.71
exception status indicator
conceptual entity that exists for each exception-name

3.72
EXIT PARAGRAPH statement
EXIT statement with the PARAGRAPH phrase

3.73
EXIT PERFORM statement
EXIT statement with the PERFORM phrase

3.74
EXIT PROGRAM statement
EXIT statement with the PROGRAM phrase

Note 1 to entry: The EXIT PROGRAM statement is an archaic feature. For details see F.1, Archaic language elements.

3.75
EXIT SECTION statement
EXIT statement with the SECTION phrase

3.76
explicit scope terminator
statement-dependent word that, by its presence, terminates the scope of that statement

3.77
extend mode
mode of file processing in which records can be added at the end of a sequential file, but no records can be deleted, read, or updated

3.78
extended letter
letter, other than the basic letters, in the set of characters defined for the COBOL character repertoire

3.79
external data
data that belongs to the run unit and can be accessed by any runtime element in which it is described

3.80
external media format
form of data suitable for presentation or printing, including any control functions necessary for representation as readable text

3.81
external switch
hardware or software device, defined and named by the implementor, that is used to indicate that one of two alternate states exists

3.82
factory object
single object associated with a class, defined by the factory definition of that class, typically used to create the instance objects of the class

3.83
file
logical entity that represents a collection of logical records

3.84
file connector
storage area that contains information about a file and is used as the linkage between a file-name and a physical file and between a file-name and its associated record area

3.85
file organization
permanent logical file structure established at the time that a file is created

3.86
file position indicator
conceptual entity that either is used to facilitate exact specification of the next record to be accessed, or indicates why such a reference cannot be established

3.87
file sharing
cooperative environment that controls concurrent access to the same physical file

3.88
fixed file attribute
attribute of a physical file that is established when the physical file is created and is never changed during the existence of the physical file

3.89
formal parameter
data-name specified in the USING phrase of the procedure division header that gives the name used in the function, method, or program for a parameter

3.90
format character
character whose primary function is to affect the layout or processing of characters around it

Note 1 to entry: A format character generally does not have a visible representation of its own.

3.91
function
intrinsic or user-defined procedural entity that returns a value based upon the arguments

3.92
function prototype definition
definition that specifies the rules governing the arguments needed for the evaluation of a particular function, the data item resulting from the evaluation of the function, and all other requirements needed for the evaluation of that function

3.93
graphic character
character, other than a control function or a format character, that has a visual representation normally handwritten, printed, or displayed

[SOURCE: ISO/IEC 10646:2020, 3.28]

3.94
graphic symbol
visual representation of a graphic character or of a composite sequence

[SOURCE: ISO/IEC 10646:2020, 3.29]

3.95
grouping
separation of digits into groups in number and currency formatting

3.96
grouping separator
character used to separate digits in numbers for ease of reading

3.97
high-order end
leftmost position of a string of characters or a string of bits

3.98
hexadecimal digit
character from the set 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F used in the representation of hexadecimal values, where the letters A-F are equivalent to the letters a-f

3.99
i-o mode
mode of file processing in which records can be read, updated, added, and deleted

3.100
i-o status
conceptual entity that exists for a file, that contains a value indicating the result of the execution of an input-output operation for that file

3.101
imperative statement
statement that specifies an unconditional action or that is a delimited scope statement

3.102
index
conceptual data item, the content of which refers to a particular element in a table

3.103
inheritance
<for classes> mechanism for using the interface and implementation of one or more classes as the basis for another class

3.104
inheritance
<for interfaces> mechanism for using the specification of one or more interfaces as the basis for another interface

3.105
initial state
state of a function, method, or program when it is first activated in a run unit

3.106
inline exception handling
facility to use PERFORM statements with the ability to trap and handle exception conditions within those statements using a WHEN phrase instead of declaratives

3.107
input mode
mode of file processing in which records can only be read

3.108
instance object
single instance of an object defined by a class and created by a factory object

3.109
interface
<of an object> names of all the methods defined for the object, including inherited methods; for each of the methods the ordered list of its formal parameters and the description and passing technique associated with each, any returned value and its description, and exceptions that can be raised

3.110
interface
<the language construct> grouping of method prototypes

3.111
internal data
all data described in a source unit except external data and external file connectors

3.112
key of reference
key, either prime or alternate, currently being used to access records within an indexed file

3.113
line delimiter
sequence of one or more bytes which terminates a record in a line sequential file

Note 1 to entry: The line delimiter is implementor-defined.

3.114
little-endian
characterized by the arrangement of data within a data item such that its most significant component occupies the highest (rightmost) component memory address within the item

3.115
locale
facility in the user's information technology environment that specifies language and cultural conventions

3.116
lock mode
state of a file for which record locking is in effect that indicates whether record locking is manual or automatic

3.117
low-order end
rightmost position of a string of characters or a string of bits

3.118
message control system
MCS
implementor-defined system that sends and receives messages exchanged between run units

3.119
message-server
run unit that receives a request via a RECEIVE statement from a requestor and returns information to that requestor

3.120
message-tag
implementor-defined unit of data that specifies the requestor or sender of a message and any additional information about the message

3.121
method invocation
invocation
request to execute a named method on a given object

3.122
method prototype
source element that specifies the information needed for invocation of a method and for conformance checking

3.123
national character position
<storage required> amount of physical storage required to store, or presentation space required to print or display, a single national character

3.124
national character position
<location within an item> location within a national data item of a national character

3.125
national character set
national coded character set
coded character set that the implementor has designated for representation of data items described as usage national and for national literals

3.126
national data item
elementary data item of class national or a national group item

3.127
native arithmetic
mode of arithmetic in which the techniques used in handling arithmetic are specified by the implementor

3.128
native character set
implementor-defined character set, either alphanumeric or national or both, that is used for internal processing of a COBOL runtime module

3.129
native collating sequence
implementor-defined collating sequence, either an alphanumeric collating sequence or a national collating sequence, that is associated with the computer on which a runtime module is executed

3.130
next record
record that logically follows the current record of a file

3.131
null
<message tag> state of a message-tag that it contains no valid message tag

3.132
null
<object reference> state of an object reference indicating that it contains no valid reference

3.133
null
<address pointer> state of an address pointer that it contains no valid address

3.134
numeric character
<in the rules of COBOL> character that belongs to the following set of digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

3.135
object
unit consisting of data and the methods that act upon that data

3.136
object data
data defined in the factory definition, except for the data defined in its methods, or in the instance definition, except for the data defined in its methods

3.137
object property
property
name that can be used to qualify an object reference to get a value from or pass a value to an object

3.138
open mode
state of a file connector indicating input-output operations that are permitted for the associated file

3.139
optional file
file declared as being not necessarily present each time the runtime module is executed

3.140
outermost program
program, together with its contained programs, that is not contained in any other program

3.141
output file
file that is opened such that it can only be written or extended

3.142
output mode
mode of file processing in which a file is created and records can only be added to the file

3.143
physical file
physical collection of physical records

3.144
previous record
record that logically precedes the current record of a file

3.145
procedure
paragraph, section, or one or more successive paragraphs or sections that include the executable code

3.146
procedure branching statement
statement that causes the explicit transfer of control to a statement other than the next executable statement in the sequence in which the statements are written

3.147
processor
computing system, both hardware and software, used for compilation of source code or execution of run units, or both

3.148
program prototype definition
definition that specifies the rules governing the class of the parameters expected to be received by a particular called program, and any other requirements needed to transfer control to and get control and return information from that called program

3.149
random access
access mode in which the value of a key data item identifies the logical record that is obtained from, deleted from, or placed into a relative or indexed file

3.150
record key
data item within a record used to identify that record within an indexed file

3.151
record lock
conceptual entity that is used to control concurrent access to a given record within a shared physical file

3.152
record locking
facility for controlling concurrent access to records in a shared physical file

3.153
relative item
item in a report whose position is specified relative to the previous item

3.154
relative key
data item that contains a relative record number

3.155
relative record number
ordinal number of a record in a file whose organization is relative

3.156
report writer
comprehensive set of data clauses and statements that enable a print layout to be described according to its general appearance rather than through of a series of procedural steps

3.157
requestor
run unit that makes a request to a message server via a SEND statement to receive information back from that message server

3.158
run unit
runtime entity consisting of one or more runtime modules that interact and that function at execution time as an independent entity

3.159
runtime element
element consisting of code and data produced by the compilation of a source element

3.160
runtime module
runtime element consisting of one or more runtime elements that result from the compilation of a compilation unit

3.161
sequential access
access method in which logical records are either placed into a file in the order of execution of the statements writing the records or obtained from a file in the sequence in which the records were written to the file

3.162
sequential file
file that is opened such that it can only be written or read sequentially

3.163
sequential organization
permanent logical file structure in which a record is identified by a predecessor-successor relationship established when the record is placed into the file

3.164
source element
source unit excluding any contained source units

3.165
source unit
sequence of statements beginning with an identification division and finishing with an end marker or the end of the compilation

Note 1 to entry: Any source units contained within a source unit are part of the containing source unit.

3.166
standard binary floating-point usages
usages float-binary-32, float-binary-64, and float-binary-128

3.167
standard decimal floating-point usages
usages float-decimal-16 and float-decimal-34

3.168
static data
data that has its last-used state when a runtime element is re-entered

3.169
subject of the entry
data item that is being defined by a data description entry

3.170
subscript
number used to refer to a specific element of a table, or in the case of the value 'ALL', to all elements of a table

3.171
superclass
class that is inherited by another class

3.172
surrogate pair
coded character representation for a single abstract character of the UTF-16 format of the UCS where the representation consists of a sequence of two two-octet values where the first value of the pair is a high-surrogate and the second is a low-surrogate

3.173
tape drive
real or virtual device that records data sequentially and once the data is written it cannot be changed

Note 1 to entry: This includes magnetic tape drives, ribbon drives, write only devices, virtual files that behave like magnetic tape drives, and similar devices.

3.174
type
template that contains all the characteristics of a data item and its subordinates

3.175
universal object reference
object reference that is not restricted to a specific class or interface

3.176
unsuccessful execution
attempted execution of a statement that does not result in the execution of all the operations specified by that statement

3.177
zero-length item
element of data whose minimum permitted length is zero and whose length at runtime is zero

3.178
zero-length literal
alphanumeric, boolean, or national literal that contains zero characters