

A

Accessing XML Documents	4:8-9
Addressing: 24 versus 31 Bit	6:3
AIXBLD	9:20
AMODE	6:4
ARITH - EXTEND or COMPAT	9:4
Assignment	2:10
Automatic Date Recognition.....	8:4
AWO or NOAWO	9:5

B

Batch Compilation	7:14-15
BINARY	9:41
BINARY (COMP or COMP-4)	9:39-40
Bit Manipulation Routines	7:45

C

CALLS	9:29-30
CEE3ABD-Terminate	7:46
CEE3DMP-Generate Dump	7:47
CEEDUMP	7:42
CEEQCEN/CEESCEN	7:35-36
CEESECS	7:37-40
Century Window	8:3
Century Window Routines.....	7:34
Century Window: Using.....	8:7
CHECK.....	9:21
CICS.....	7:17
Class - Defining	3:15
Class in Object-oriented COBOL	3:9
CLASS-ID Paragraph.....	3:16-17
COBOL Data Type Definitions	7:43-44
COBOL Facilities.....	4:10
COBOL in OO (Using).....	3:6
COBOL OO Features.....	3:4
COBOL Unicode Support and XML Processing	2:15
COBOL XML Parser Support.....	1:13-15
CODEPAGE	4:14
Coding Attribute (get and set) Methods	3:36
COMMON.....	5:11
COMP-5	9:42
Compiler	7:3
Compiler Options	2:9
Compiler Options that Affect Run-Time Performance	9:3
Compiler Options Under z/OS	7:8
Content of XML-CODE.....	4:21
Control Run Time Storage	6:6-7
Creating and Initializing Instances of Classes	3:38
CURRENT-DATE	8:14-17

D

Data Conversions.....	9:43
DATA DIVISION for Defining a Class Instance Method	3:26
DATA(24) or DATA(31)	9:6
Date and Time Service.....	7:28
Date and Time Values: Performing Calculations	7:33
Date Format	7:32
Date/Time Callable Services	7:29-31
DATE-OF-INTEGERS	8:18
Date-related Diagnostic Messages	8:13
Date-related Logic Problems.....	8:6
DATE-TO-YYYYMMDD	8:19
DATEVAL	8:11, 8:20
DAY-OF-INTEGERS	8:21
DD Cards for Compile	7:4-7
DEBUG.....	9:22
Defining a Class	3:15
Defining a Class Instance Method	3:21-22
DISPLAY	9:44
Display.....	9:45
DYNAM or NODYNAM.....	9:7

E

EBCDIC Code Pages - Supported	4:25
Elementary Items	4:38
EXCEPTION	4:16

F

Factory Data, Static Data Used in Object-oriented COBOL	3:13
Factory Method, Static Method Used in Object-oriented COBOL	3:12
FASTSORT or NOSORT	9:8
Filler.....	4:37
First Program Not COBOL	9:35
Fixed-Point versus Floating-Point	9:47
Freeing Instances of Classes.....	3:39
Full Field Expansion: Moving to	8:9-10

G

General Callable Services	7:41
Generating XML Output	4:35
Global Names	5:14

H

HEAP.....	6:10
-----------	------

I

Indexes versus Subscripts	9:48
Inheritance Used in Object-oriented COBOL	3:14
Initial	5:16
Initial State	5:18
INPUT-OUTPUT SECTION for Defining a Class Instance Method	3:25
Instance Data Used in Object-oriented COBOL	3:11
Instance Method - Overloading	3:35
Instance Method Used in Object-oriented COBOL	3:10
INTEGER-OF-DATE	8:22
INTEGER-PART	8:23
Internal Bridging: Using	8:8
Intrinsic Conversion Functions	2:13
Invoking Methods (INVOKE)	3:37
IS INITIAL on the PROGRAM-ID Statement	9:32-33
IS RECURSIVE on the PROGRAM-ID Statement	9:34

J

Java Interoperability	1:5, 1:7
JNI: Java Native Interface	3:8

L

Language Environment and Run Time Options	7:18
Language Environment Initialization	7:24-26
LE Date and Time	7:27
Link-edit Considerations	4:11
List of Compiler Error Messages	7:16
Local Names	5:13
LOCAL-STORAGE SECTION	3:27
Location of Data Areas	6:9

M

METHOD-ID	3:24
Migration Path	1:8
Miscellaneous Enhancements	1:22
MLE: Millennium Language Extensions	8:5
Multithreading	1:20-21

N

Nested Program	5:7-8
Nesting Called Programs: Advantages	5:9
New Features	1:3-4
NUMPROC - NOPFD, MIG, or PFD	9:9-10

O

OCCURS DEPENDING ON	9:49
OO Program	3:3, 3:5
OO Syntax for Java Interoperation	1:6
OO Terms	3:7
OPTIMIZE(STD), OPTIMIZE(FULL), or NOOPTIMIZE	9:11
Other Language Syntax Supporting Unicode	2:12
Overloading an Instance Method	3:35

P

PACKED-DECIMAL (COMP-3).....	9:46
Parser Control	4:17
Parsing XML Documents	4:12
PROCEDURE DIVISION: Defining a Class Instance Method	3:29-30
PROCESS (CBL) Statement	7:9
Processing XML Input.....	4:3
Program Pretty	4:45-49
Program Structure.....	7:23

Q

QSAM Files	9:31, 9:36
------------------	------------

R

Recursive Calls	5:3
Recursive Calls: Making.....	5:5
Recursive: Identifying a Program	5:4
Redefines	4:36
REPOSITORY Paragraph for Defining a Class	3:18-19
RMODE	6:5
RPTOPTS	9:23
RPTSTG	9:24
Run Time Options	7:19-22
Run-Time Options that Affect Run-Time Performance	9:19

S

Sample Class	3:31
Sample Client.....	3:40
Sample Factory	3:44-49
Sample INITIAL.....	5:19
Sample Method	3:32-34
Sample Nested Program.....	5:10
Sample Parse.....	4:27-32
Sample Recursive	5:6
Sample SubClass.....	3:41-43
Sample XML Output Program.....	4:39-44
Scope of Names.....	5:12
Searches for Name Declarations	5:15
Setting a Program to an Initial State	5:17
Signature	3:23
Special Registers	4:4, 4:20
SSRANGE or NOSSRANGE	9:12
Storage for LOCAL-STORAGE Data.....	6:11
Storage Management Tuning.....	9:26-28
Storage Restrictions for Passing Data.....	6:8
Symbolic Characters	5:23-24

I

TEST	9:25
TEST or NOTEST	9:13
THREAD or NOTHREAD	9:14
TITLE Statement	5:20, 5:22
Title.....	5:21
Transforming XML Text to COBOL.....	4:23-24
TRUNC - BIN, STD, or OPT.....	9:15-18
Types of Compiler Output Under z/OS	7:10-13

U

UNDATE.....	8:12, 8:24
Unicode	1:16-18
Unicode Compares	2:11
Unicode Conversion.....	1:19
Unicode Data Type	2:8
Unicode Example	2:16
Unicode Literals	2:7
Unicode Support in Enterprise COBOL for z/OS	2:5
Unicode Support: Overview	2:6
Unicode: Using in DB2 COBOL Programs.....	2:14
Unicode: What is it?	2:3
Unicode: Why.....	2:4

V

VSAM	9:38
VSAM Files.....	9:37

W

WebSphere Support.....	1:9
Well Formed.....	4:7
WORKING-STORAGE SECTION.....	3:28
WORKING-STORAGE SECTION for Defining Class Instance Data.....	3:20

X

XML Declaration.....	4:15
XML Document - Sample.....	1:11
XML Documents - Accessing.....	4:8-9
XML-EVENT.....	4:18
XML Input - Processing.....	4:3
XML: Introduction	1:10
XML on z/OS.....	1:12
XML Output	4:33-34
XML Output - Generating.....	4:35
XML Parse Processing.....	4:6
XML PARSE.....	4:13
XML Parser - Handles Errors.....	4:26
XML Parser in COBOL.....	4:5
XML-TEXT and XML-NTEXT	4:22
XML Writing Procedures to Process XML Events	4:19

Y

YEAR-TO-YYYY	8:25
YEARWINDOW.....	8:26