COBOL for Programmers (All Versions)

Duration

5 to 8 days.

Participants

Experienced programmers who will be responsible for developing and maintaining COBOL application programs.

Overview

This course covers the topics in our Entry Level COBOL course except those dealing with program logic, which students will already know from their experience in another application development language.

Objectives

After successfully completing this course, you will be able to:

- Write moderately complex COBOL programs that use efficient coding techniques, are easily maintained and modified, and exploit the structured capabilities in COBOL II and higher releases.
- Debug programs using COBOL's facilities for debugging abends and logic errors.
- Code JCL compiler options best suited to the program and operating system.
- *(Optional - One day.)* Identify differences between OS/ VS COBOL and VS COBOL II in order to convert OS/ VS COBOL programs to COBOL II.

Format

Lecture and hands-on computer workshops. Throughout the course, students perform many hands-on workshops and coding exercises to reinforce material presented in lectures. The student text includes many COBOL coding examples.

Variations of this course include presentation on the Micro Focus COBOL Workbench instead of a mainframe, and presenting different dialects, such as COBOL/ 370 (now called COBOL for MVS and VM), OS/ VS COBOL, or PC COBOL.

For participants without a programming background, we suggest our Entry-Level COBOL Programming class, which includes JCL, logic, ISPF or equivalent, utilities, tools, and application development methodologies..

Prerequisites

Programming experience in a procedural language such as C, C++, PL/ I, Assembler, and so on is necessary to complete the course in five days. Participants must know procedural logic as this is not presented in this course. Additional time is recommended if students have limited programming experience. Knowledge of basic MVS JCL and ISPF is required, if presented on MVS.

Topic Outline

Language Structure

Coding Format and Rules
Writing Structured COBOL Programs
Documentation

Basic Input/Output

ACCEPT, DISPLAY
JCL

Topic Outline

Data Handling

- Data Description
  - PICTURE, USING, VALUE
  - COMP, COMP-3, Binary, Packed Decimal
  - DISPLAY
- Data Manipulation
  - MOVE
  - MOVE. . . CORR
  - SET
- Arithmetic Operations
  - ADD, SUBTRACT, MULTIPLY, DIVIDE
  - COMPUTE (GIVING, ROUNDED, ON SIZE ERROR)
- Dates and date processing
  - Year 2000 issues

Program Control

- STOP RUN, GOBACK
- Scope terminators
- PERFORM (THRU, UNTIL, TIMES, VARYING)
- EXIT
- IF, EVALUATE
- CONTINUE, NEXT SENTENCE
- GO TO (DEPENDING ON, Level-88)

Input/Output Processing

- File Definitions
- Record Descriptions
- Sequential Processing
- Print File Control Characters
- I/ O Verbs
  - READ, WRITE
  - OPEN, CLOSE
  - AFTER/ BEFORE ADVANCING

Creating and Processing Tables

- Definition in DATA DIVISION
- One-dimensional tables
- PERFORM formats: COBOL vs COBOL II
- Comparison of indexing and subscripting
  - Location in main storage
  - Initializing and altering values
  - Differences in processing efficiency
  - Debugging differences

Topic Outline

Creating and Processing Tables *(continued)*

- Techniques to initialize tables
  - REDEFINES
  - VALUE on group item
  - Calculate values
  - Load values from file
  - Initializing tables at compile time or execution time
- Processing tables
  - One occurrence, section of table, entire table
  - Serial search
  - Binary search
- Handling out-of-bounds situations

Implementing Table Processing

- Use of SET
- Processing one-dimensional tables
- Processing variable-length and large tables
  - Definition in DATA DIVISION
  - Variable-length tables
    - Concept
    - Defining the length (OCCURS . . . DEPENDING ON . . .)
    - Program logic to load and process table
    - Processing V, VB, and VBS records
  - Large table handling techniques
    - Table limitations
    - Breaking up a table
    - Disk tables
  - USAGE IS INDEXED
- Processing one-dimensional variable-length tables
- Multidimensional tables
  - Definition in DATA DIVISION
  - Processing two-dimensional tables
  - Combining one, two, and three dimensions in a table
    - Definition in DATA DIVISION
    - Processing considerations

Subprogram Linkage

- CALL statement
- Passing parameters
- LINKAGE SECTION
- STOP RUN vs GOBACK
- CANCEL
- Static vs dynamic calls
- Reentrant programs
- Multiple entry points

Topic Outline

Subprogram Linkage ***(continued)***

- Running below or above the 16-megabyte line:
  - DATA(24) and DATA(31)
  - AMODE and RMODE
- Basic Linkage Editor control statements
- Linking from object or load modules
- Passing data vs passing addresses
  - CALL BY CONTENT / CALL BY REFERENCE
- Passing data from the JCL
- Nested programs

Implementing Subprogram Linkage

Additional COBOL Verbs and Capabilities

- SORT
  - SD, USING, GIVING
  - INPUT PROCEDURE, OUTPUT PROCEDURE
- MERGE
- Reference Modification (Substring)
- Hexadecimal
- STRING
- UNSTRING
- INSPECT
- Pointers

Converting Programs from COBOL to COBOL II ***(Optional)***

Testing and Debugging
(If taught using Micro Focus COBOL Workbench, then the Animator V2 debugging capabilities are presented instead.)

- COBOL Debugging Facilities
  - FDUMP, OFFSET, LIST, MAP
  - READY/ RESET TRACE
- Compiler Output and Debugging
  - Procedure Division Maps
    - Condensed Listing
    - Assembler Listing
  - Data Division Map
  - COBOL Work Areas
  - Locating Abending Statement
  - Locating Data Areas in Storage
- RENT vs NORENT impact