

# Entry Level COBOL Programming

---

<b>Duration</b>	25 days minimum <i>(See timings for each module.)</i>
<b>Participants</b>	Anyone with an interest in and an aptitude in being a COBOL programmer. Limited or no programming experience is assumed, unless otherwise specified by the client. We recommend that each student in the class have a similar background. (We can and have presented this course many times to experienced C, C++, Assembler, Java, and PL/ I programmers.).
<b>Objectives</b>	<p>Upon successful completion of this course, you will be able to:</p> <ul style="list-style-type: none"><li>• Write moderately complex COBOL programs that use efficient coding techniques, are easily maintained and modified, and exploit COBOL's structured and function capabilities.</li><li>• Implement a structured program using top-down modular design and the fundamental logic structures.</li><li>• Use tools such as functional decomposition (hierarchy charts/ action diagrams) and pseudocode to analyze and model the logic of a program.</li><li>• Test and debug COBOL programs using COBOL's facilities for debugging abends and logic errors and appropriate testing techniques.</li><li>• Code JCL compiler options best suited to the program and operating environment.</li></ul>
<b>Prerequisites</b>	Limited or no programming experience is assumed, unless otherwise specified by the client. We recommend that each student in the class have a similar background.
<b>Overview</b>	<p>A series of modules provides the participant with all the tools necessary to begin a programming career in COBOL for z/ OS or COBOL for MVS and VM. At the end of this series of modules, participants can program moderately difficult programs and integrate them with the rest of the system. Graduates of this program are an asset to any program team.</p> <p>The first module in this series is an overview of computing concepts, terminology, and the major software tools, operating system features, and equipment available to the programmer - the computer components. The next two modules show the student how to allocate these components - devices, memory, time, programs. Then the focus turns to application development.</p> <p>For example, the modules can include the following in your curriculum:</p> <ul style="list-style-type: none"><li>• z/ OS or OS/ 390 Mainframe Computer Concepts (1 day)</li><li>• MVS and TSO/ ISPF (3 days) (or VM and CMS)</li><li>• JCL and Utilities (5 days) (Includes IEBGENER, IDCAMS, and SORT.)</li></ul>

*(continued)*

# Entry Level COBOL Programming *(continued)*

---

## Overview *(continued)*

- COBOL Programming and Logic (15 to 25\*\* days) (Enterprise COBOL for z/ OS, COBOL/ 390, or COBOL for MVS and VM.)
- VSAM and related IDCAMS (5 days)
- QMF Workshop (3 days) (Optional\*\*\*\*)
- Additional tools, such as FileAid, CompareX, Xpediter, etc. (Optional, 1 day for each course.)
- DB2/ SQL Application Development (5 to 10\* days)
- CICS Command Level Programming (5 to 12\* days)
- Case study \*\*\* (1 to 6\* days)

*Total length ranges from 40 to 70 days*

\* When we present subjects such as VSAM, DB2, IMS, and CICS in an entry level curriculum, we focus on the procedural logic required as well as on the syntax and efficiencies of each programming tool. Therefore, the lengths are extended to include logic lectures and exercises, as well as to reinforce student's abilities in these areas. IMS DB, DC (TM), and MFS may be substituted for DB2 and/ or CICS.

\*\* By extending this course to 25 days, a number of advanced topics can be presented, and structured programming design and logic skills can be solidified.

\*\*\* We recommend concluding this course with a case study workshop that brings together all the material that students have learned in this program. This case can be customized to the client's environment.

\*\*\*\* To truly strengthen basic DB2 skills, QMF can also be presented (3 days).

## Format

Lecture and hands-on computer workshops. This course takes you from design to finished code focusing on structure in every step of the process from structured design through structured coding and testing, including the use of TSO structured facilities. In the workshops, you design, code, test, and debug a variety of programs that use logic structures fundamental to business applications. (VM and CMS concepts and usage may be substituted for the MVS and ISPF/ TSO module. Both may be presented by adding an additional 2 days.)