

# JCL for COBOL Programmers

CDT030

This course is a follow-up to our [JCL by Example](#). It picks up where that course left off, with topics of special interest to COBOL programmers and more sophisticated users of JCL. Topics include the COBOL compile and link process, STEPLIB vs. JOBLIB, accessing PARM data, generation data groups (GDGs), variable length records, and more work with catalogued procedures (PROCs). As with the first course, there is plenty of hands-on time. *This course is particularly useful to new mainframe programmers who used PC-based compilers in school. Note: participants must have some COBOL experience or education!*

---

## Audience

- Mainframe COBOL Programmers

## Prerequisites

- COBOL experience or education is required
- Knowledge of TSO/ISPF and JCL (preferably completion of our [JCL by Example](#))

## Teaching Methods

- Lecture with examples
- Supplemental hands-on exercises

## Course Length

- Two Days
- 

## Course Outline

QE0

### COBOL compile and link

- Allocating source, copy, and load libraries
- COBOL compile and link JCL
- Using a symbolic in the compile and link JCL

### STEPLIB vs. JOBLIB

- Accessing other load libraries
- S806 abend

### Accessing PARM data

- Defining the PARM data in the LINKAGE SECTION
- Accessing the PARM data in the PROCEDURE DIVISION
- Validating PARM data

### User abends

- RETURN-CODE
- Forcing a user abend

### Generation data groups

- Defining the GDG base
- Creating the model DSCB
- Creating the first generation
- Creating subsequent generations
- LISTCAT
- Mass access to all generations
- Deleting the GDG base
- Case Study: Small Town Hardware Store Daily Processing

### Variable length records

- Why variable length records
- RECFM=VB
- Record descriptor word (RDW)
- Block descriptor word (BDW)
- COBOL's OCCURS DEPENDING ON clause

### More PROC practice

- A permissive delete PROC
- A PROC for the Small Town Hardware Store
- Daily Processing