Assembler Programming Part I

Duration: 5 Days

Audience:

Application Programmers with little or no previous experience in Assembler who require formal training in the basic language features and programming techniques.

Pre-requisites:

An understanding of computer concepts is assumed.

A working knowledge of TSO/ISPF is required. This can be gained from our z/OS TSO/ISPF Workshop.

Course Objectives

Each delegate will acquire a working knowledge of Assembler and will gain a solid foundation in the fundamentals of Assembler coding including program structure, design, execution and debugging. Good programming practice is encouraged throughout. The course starts with the basics and furthers learning with 54 hands on assignments until delegates are capable of complex programming logic and design.

Course Content

Module 1: Documentation

A variety of manuals is presented to aid coding. An explanation of the Assembly process is also covered.

Module 2: Hardware Appreciation

The structure of memory
AMODE vs RMODE
How programs and data are located
z/Architecture registers (3-types)
Register conventions
Program Status Word (PSW)

Module 3: Getting Started

General syntax
PRINT, TITLE, CSECT and START
Comments

Naming a Control Section (CSECT)
Gaining control – STM, STMG, BAKR, USING, BASR, ST, STG and LA instructions
Returning control – PR, L, LG, LM, LMG and BR instructions, plus the SAVE macro
Setting a Return Code
The END statement

Module 4: Data Area Definition

Declare Constant (DC) statement
Types of constant – Address, Binary, Character, Decimal, Hexadecimal etc.
Data attributes – Duplication factor, Initial value and length
Data padding and Truncation
Define Storage (DS) statement
Data attributes – Duplication factor, Initial value and length
Simulating Group Structures
COPY statement
Literals and the LTORG statement
The EQU statement
Where to place data items

Module 5: Relative Addressing

Base Displacement revisited
Multiple base registers
SYSSTATE Macro
IEABRC and IEABRCX Macros
Establish Static Area Addressability – LARL and STRL instructions
Qualified USING statements
Potential issues and the need for local addressability

Module 6: Diagnostic Aids

Checking logic flow – WTO Macro
Forcing an ABEND – ABEND Macro
Checking the contents of memory or registers – SNAP Macro

Module 7: File Processing (QSAM)

Access Methods – BSAM vs QSAM
OPEN Macro
GET Macro – both Move and Locate modes
PUT Macro – both Move and Locate modes

CLOSE Macro
DCB Macro – both input and output
DCBE Macro – required for 31-bit I/O processing

Module 8: Decision Making

PSW Condition Code
Branching - BC, BCR and BRC Instructions
Character compares – CLC, CLI and CLCL instructions
Binary compares – C, CG, CGH, CGR, CH and CR instructions
Iterative branching – BCT, BCTG and BCTGR and BCTR instructions
Extended mnemonics

Module 9: Using Memory

Move/Copy multiple characters – MVC, MVCL and MVST instructions
Move/Copy a single byte – MVI instruction
Move/Copy half a byte – MVN and MVZ instructions
Move/Copy shifting left 4-bits – MVO instruction
Handling variable length moves – EX instruction

Module 10: Using Registers

Memory to Register – IC, ICY, L, LA, LGA, and LH instructions
Register to Register – LR, LGR, LTR and LTGR instructions
Register to Memory – STC, STCY and STH instructions
Memory to Register using a mask – ICM and ICMH instructions
Register to Memory using a mask – STCM and STCMH instructions

Module 11: DSECT and ORG statements

DSECT statement
DSECT structure
Ending a DSECT
ORG statement
Indicating the end of an ORG sequence

Module 12: Data conversion

Why conversion is necessary
Convert text to decimal – PACK instruction
Convert decimal to binary – CVB and CVBG instructions
Convert binary to decimal – CVD and CVDG instructions
Convert decimal to text – UNPK instruction
Decimal signs explained

Translate a character set – TR instruction
Search for a character – TRT instruction

Module 13: Binary Integer Arithmetic

Addition – A, AG, AGR, AH, AHY and AR instructions
Division – D and DR instructions
Multiplication – M, MH and MR instructions
Subtraction – S, SG, SGR, SH, SHY and SR instructions
Binary search example

Module 14: Decimal Arithmetic

Addition – AP and ZAP instructions
Division – DP instruction
Multiplication – MP instruction
Subtraction – SP instruction
Compare – CP instruction
Performing rounding – SRP instruction
Display results – ED and EDMK instructions
Edit patterns explained

Module 15: VSAM I/O Macros

Emphasis on Entry Sequenced Data Sets (ESDS)
Cluster processing – OPEN, GET, PUT and CLOSE Macros
File definition – ACB, EXLST and RPL Macros
Sundry other macros – ENDREQ, GENCB, MODCB and SHOWCB

Appendix A: Linkage Editor / Binder

Overview
Primary input – SYSLIN
Primary output – SYSLMOD
INCLUDE statement
NAME statement
List of other statements
PARM field operands

Appendix B: EXITS (a brief description)

What are they?
What exploits them?
Where will the exit execute?
Re-entrancy requirements
Obtaining the re-entrant attribute
The STORAGE macro
Avoiding in-line parameter lists
RSECT vs CSECT
Link-Edit / Program Binder requirements