

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 enhance their working knowledge of Assembler as this course builds on the topics from Assembler Programming Part I. Good programming practice is encouraged throughout. The course covers a range of topics useful in creating more complex programs and furthers learning with 40 hands on assignments until delegates are capable of complex programming logic and design.

Course Content

Module 1: Assembler Programming Part I revision

AMODE vs RMODE plus a list of instructions

Module 2: Diagnostic Aids

Types of termination Forcing an SOC1 Abend More information on the SNAP macro Forcing a User ABEND (U0001-U3999) Trapping program interrupts – SPIE and ESPIE macros Trapping System Abends – STAE, ESTAE and ESTAEX macros SETRP Macro

Module 3: More Decision Making

Why return codes are multiples of 4 Branch tables EX instruction



Branch on Index High – BXH instruction Branch on Index Low or Equal – BXLE instruction

Module 4: Static Sub-routines

Retrieving information from the EXEC statement PARM operand Internal sub-routines – BAL, BALR, BAS, BASR and BR instructions External sub-routines – CALL macro How to return from an external sub-routine Passing parameters to an external sub-routine The role of the Linkage Editor / Program Binder Linkage Editor DD statements Common Linkage Editor statements – ENTRY, INCLUDE and NAME Common Linkage Editor PARM options – LET, LIST, MAP and NCAL

Module 5: Dynamic Sub-routines

The differences between CALL, LINK, LOAD and XCTL macros Dynamic invocation with return – CALL, LINK and LOAD macros Dynamic invocation without return – XCTL macro DELETE Macro Changing AMODE in-flight

Module 6: Sub-Tasks

Concepts of sub-tasking Creating a sub-task – ATTACH and ATTACHX macros Deleting a sub-task – DETACH macro STATUS Macro Synchronizing tasks – ECB, POST and WAIT macros CS Instruction

Module 7: Boolean Instructions

Instruction types Basic definitions – AND, OR and Exclusive OR Register to Register (RR-type) – NGR, NR, OGR, OR, XGR and XR instructions Register to Memory (RX-type) – NG, N, OG, O, XG and X instructions Immediate instructions (SI-type) – NI, OI, and XI instructions Storage to Storage (SS-type) – NC, OC and XC instructions Checking bit settings – TM instruction



Module 8: Shift Instructions

Single shift arithmetic instructions Single shift logical instructions Double shift arithmetic instructions Double shift logical instructions

Module 9: BSAM and BPAM I/O

Access Methods BSAM vs QSAM OPEN macro Read a block of data – READ macro Write a block of data – WRITE macro Handling short blocks both in and out Validating the I/O – CHECK macro CLOSE macro Defining the data set – DCB Macro (both input and output) Additional BPAM requirements – FIND and STOW macros

Module 10: VSAM Processing

Cluster processing – OPEN, GET, PUT and CLOSE Macros File definition – ACB, EXLST and RPL Macros Sundry other macros – ENDREQ, GENCB, MODCB and SHOWCB

Module 11: Memory Management

What does it mean What is a sub-pool Acquiring areas of memory – GETMAIN and STORAGE macros Releasing areas of memory – FREEMAIN and STORAGE macros

Module 12: Language Environment (LE)

Register conventions Mandatory macros – CEECAA, CEEDSA, CEEENTRY, CEEPPA, and CEETERM Additional macros – CEEFETCH, CEELOAD, CEEPCAL and CEERELES AMODE and RMODE implications

Module 13: Macro Basic Concepts

Naming the macro – MACRO statement Defining the macro attributes – Prototype statement Creating the macro logic – Model statements



Terminating the macro – MEND statement Issuing messages from the macro – MNOTE statement Aborting a macro – MEXIT statement Additional statements – AEJECT, AINSERT, AREAD, ASPACE and COPY Macro placement

Module 14: Variable use in Macros

Symbolic parameter both Keyword and Positional Sub-parameter lists Symbolic parameter concatenation Available System Symbols Significance of special characters

Module 15: Conditional Assembly

Its functionality and components Declaring local symbols – LCLA, LCLB and LCLC statements Declaring global symbols – GBLA, GBLB and GBLC statements Assigning values – SETA, SETB and SETC statements Sequence symbols Branching – AIF, AGO and ANOP statements Iterative processing – ACTR statement

Module 16: Data Spaces

Data Space vs Hiperspace Access Registers Access Lists Setting an Access Register – CPYA, LAE, LAM and SAR instructions Create a Data Space Make the Data Space Addressable Populate the Data Space Using macros in AR mode – SYSSTATE macro Deleting a Data Space Other Data Space management options

Module 17: Re-Entrant Code

What is re-entrant code and what are its benefits Avoid in-line parameter lists generated by macros Use external data areas – STORAGE macro Let the assembler check for validity – RSECT statement