

# Assembler Language Programming – Basic

---

|                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Duration</b>      | 5 days                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Participants</b>  | This course is designed for the beginner assembler language application programmer who needs to code, maintain, and/ or debug application programs written in zSeries, System/ 390, or System/ 370 Assembler Language.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Objectives</b>    | <p>Upon successful completion of this course, you will be able to:</p> <ul style="list-style-type: none"><li>• Describe the steps used in the assembly process</li><li>• Entry and return macros and other standards for coding assembler programs.</li><li>• Code in a recommended program organization that includes modular routines, standard register usage, DSECTs, ORGs that are self-adjusting for length changes, and well-documented maintainable code.</li><li>• Define and use assembler pseudo-instructions.</li><li>• Code all the macros for processing QSAM files (physical sequential files) using both Move mode and Locate mode.</li><li>• Code the instructions for compare and branch looping.</li><li>• Code data validation routines that check for valid input data, including numeric validation.</li><li>• Code report programs that use editing to create formatted output fields.</li><li>• Code assembler language programs that use: standard linkage conventions, data area and constant definitions, decimal arithmetic operations, data validation, data movement operations, and typical SS instructions, including TR, TRT, MVCL, CLCL, and SRP.</li><li>• Use multiple base registers and literal control for managing large programs.</li><li>• Code "above the line" assembler language programs that use 31-bit addressing and describe the instructions that are restricted to 24-bit addressing.</li><li>• Debug assembler language programs using SNAP dumps and system-provided storage dumps and standard debugging techniques.</li><li>• Use Principles of Operation and other IBM books available in Book Manager as well as those in QuickRef.</li></ul> |
| <b>Format</b>        | Lecture and hands-on workshops.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Prerequisites</b> | <p>Before taking this course, you should be able to:</p> <ul style="list-style-type: none"><li>• Program in a procedural language, such as COBOL.</li><li>• Understand basic data processing concepts and terminology.</li><li>• Code basic JCL statements.</li><li>• Understand programming fundamentals, such as data manipulation, looping, conditional processing, and I/ O.</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

# Assembler Language Programming – Basic *(continued)*

---

**Overview** This course teaches how to use the zSeries, System/ 370, or System/ 390 Assembler Language instruction set. It includes standards for coding assembler programs, discussions of the base-displacement technique of addressing main storage, use of hexadecimal notation for identifying storage addresses and contents, coding rules and standards, the assembly process, standard QSAM I/ O macros, "above and below the line" coding considerations, and basic dump reading. Emphasis is placed on modular and structured programs and register use to facilitate debugging and dump reading.

**Topic Outline**      **Basic Assembler Language Programming**

What is Assembler?  
Reference Books and Manuals

**Prepare and Run Assembler Programs**

Sample Program  
Assembler Syntax  
Rules for Labels  
Program Preparation

**Numbering Systems and Data Storage**

Main Storage and AMODE  
Concepts of Bases  
Binary  
Hexadecimal  
Converting Between Hexadecimal and Binary  
Converting Numbers  
What Does The Hex Mean?  
Data Representation  
Data Sizes  
Character Data  
Zoned Decimal  
Packed Decimal  
Pack a Number  
Packed Decimal Data  
Unpack a Packed Decimal Number  
Fixed Binary Point  
Numeric Edited  
Hexadecimal Arithmetic  
IBM Mainframe Data Forms  
Converting Between Number Systems  
Strongly vs Weakly Typed Data  
Define Storage, Define Constant  
DS and DC Location  
DC Continuation  
Truncation and Padding  
DC Storage Types  
DS / DC Structures and ORG

**Displacement Concepts**

VSAM File Organization  
KSDS

# Assembler Language Programming – Basic *(continued)*

---

## Topic Outline

### Base Displacement Concepts

- General Purpose Registers
- GPR Typical Usage
- Base Displacement Concept
- Establish Base Register
- Labels Translate to Base Displacement
- Machine Instruction Formats
- Review: Machine Instruction Format
- RR Machine Instruction Format
- SS Machine Instruction Format

### Program Logic: Arithmetic and Modularization

- Entry and Exit Linkage
- Arithmetic Sequence for Raw Data
- Arithmetic Sequence for Stored Data
- AP and SP
- Branch Instructions
- Structured Code Using Multiple Registers
- Structured Code Using Just One Register

### Debugging Tips

- Simple Debugging Tips
- Debugging S0Cn Abends
- More Simple Debugging Tips
- Eye Catcher Literal
- SNAP Macro

### Moving Data

- Moving Data Concepts
- Moving Data
- MVC Command
- MVI Command
- Tips for Handling Negatives
- Zone vs Numeric; MVZ and MVN

### More Packed Instructions

- Self-Defining Constants and LTORG
- ORG to Extend a Record
- ORG to Redefine a Field
- ORG for VB Records
- Self-Defining Constants vs SI
- ZAP
- Multiply Packed Concept
- Divide Packed Concept
- Truncating Packed Numbers
- Decimal Places and MVN
- Decimal Places and MVO
- Rounding - SRPVSAM File Organization

# Assembler Language Programming – Basic *(continued)*

---

## Topic Outline

### Compare Values and Branch

- Compares
- Branch After Compare
- Branch After Arithmetic
- RX Instructions

### Editing Numbers and Printing Records

- Types of Instructions
- Numeric Editing (ED)
- Common Editing Pattern Characteristics
- LA and BCTR
- EDMK
- Fancy Equates for Lengths
- Printing Reports
- Printing Checklist

### Fixed-Point Binary Instructions

- FPB Logic for Processing
- Binary Instruction Set Introduction
- Benefits and Limitation
- FPB Typical Sequence
- FPB Multiply
- FPB Divide
- Load and Test Register
- Insert Character

### Basic Table Processing

- BCT LOOPS with TABLES - Packed Decimal
- BCT LOOPS with TABLES – Fixed-Point Binary
- BCT Loop with LTR
- Advanced Looping Instructions

### Program Sectioning and DSECTS

- Multiple Base Registers
- DSECT
- Get and Put Locate with DSECT
- Put Locate

### TR, TRT, and LA Instructions

- Load Address (LA)
- Translate (TR) - To Convert Bytes to Different Bytes
- Translate (TR) - To Convert To Lowercase
- Translate and Test (TRT) - To Test Bytes
- Numeric Test
- Fancy Lengths Examples
- List of Extended Mnemonics Available