Faysal Aziz

ABAP Programming Foundations

From Syntax to Algorithms in 3 days

FA-0022 · 3 days

Software Development

Introduction

Master core ABAP programming concepts through hands-on exercises and structured problem-solving.

This course focuses on building solid ABAP programming fundamentals through progressive, hands-on exercises. You'll start with basic syntax and data types, then advance through data structures, algorithms, and modern ABAP programming patterns.

Each concept is reinforced through practical coding exercises that mirror real-world programming challenges. By the final day, you'll have mastered the core programming skills that form the foundation of professional ABAP development, with a portfolio of solved problems demonstrating your capabilities.

Learning outcomes

By the end of this course, participants will be able to:

- Write syntactically correct ABAP programs with proper structure, comments, and formatting conventions.
- Declare and manipulate all major ABAP data types including strings, numbers, and complex data structures.
- Create, populate, and manipulate internal tables using various operations including sorting, filtering, and aggregation.
- Implement control flow structures including conditional statements, loops, and nested logic patterns.
- Apply algorithmic thinking to solve complex programming problems using ABAP.
- Use modern ABAP syntax features including inline declarations, chained statements, and expressions.

- Design modular, readable code following clean coding principles and best practices.
- Debug and optimize ABAP programs for performance and maintainability.

Prerequisites

- Proficiency in any modern procedural or object-oriented programming language (e.g., Java, Python, C#).
- Basic understanding of programming concepts: variables, loops, conditionals, functions.
- Familiarity with algorithmic thinking and problem-solving approaches.

Training outline

1. ABAP Programming Fundamentals

Introduction to ABAP Syntax

1. Program structure: REPORT statements, basic program layout
2. Statement syntax: period termination, case sensitivity, whitespace rules
3. Comments and documentation: asterisk (*) and quote (") comment styles

Data Types and Variable Declaration

1. Elementary types: I (integer), P (packed), F (float), C (character), STRING
2. Variable declaration: DATA statements and type specifications
3. Inline declarations: DATA(...) syntax for modern ABAP
4. Constants and literals: CONSTANTS declaration and usage

Basic Input and Output

1. WRITE statements for output formatting
2. String manipulation and concatenation
3. Number formatting and display options

Control Flow Fundamentals

1. Conditional logic: IF/ELSEIF/ELSE structures
2. CASE statements for multi-branch logic
3. Logical operators and boolean expressions

2. Data Structures and Manipulation

Internal Tables Fundamentals

1. Table types: standard, sorted, hashed tables

2. Table declaration: TYPES and DATA statements for tables
3. Work areas and header lines

Internal Table Operations

1. Populating tables: APPEND, INSERT statements
2. Reading data: READ TABLE, LOOP AT structures
3. Modifying entries: MODIFY, DELETE operations
4. Table processing: SORT, searching, and filtering

Advanced Data Structures

1. Nested internal tables and complex structures
2. Multi-dimensional data handling
3. Data aggregation and grouping techniques

String Processing

1. String operations: concatenation, splitting, replacement
2. Pattern matching and text analysis
3. Character manipulation and encoding

3. Algorithms and Problem Solving

Iterative Processing

1. LOOP constructs: LOOP AT, WHILE, DO loops
2. Loop control: CONTINUE, EXIT statements
3. Nested loops and performance considerations

Mathematical Algorithms

1. Arithmetic operations and mathematical functions
2. Number theory problems: prime numbers, factorization
3. Sequence generation and mathematical series

Text Processing Algorithms

1. String analysis: anagrams, palindromes, character counting
2. Text transformation: encoding, decoding, cipher algorithms
3. Pattern recognition and string matching

Data Processing Patterns

1. Sorting algorithms and custom sort criteria
2. Search algorithms: linear search, binary search concepts
3. Data validation and error detection

4. Advanced Programming Concepts

Modular Programming Principles

1. Code organization and structure
2. Reusable code patterns and techniques
3. Parameter passing and data encapsulation concepts

Modern ABAP Syntax

1. Chained statements and operator precedence
2. Inline expressions and functional programming concepts
3. Advanced string templates and formatting

Performance and Optimization

1. Efficient internal table operations
2. Memory management and data structure selection
3. Algorithm complexity and performance analysis

Code Quality and Best Practices

1. Readable code structure and naming conventions
2. Error handling patterns and defensive programming
3. Testing approaches and validation techniques

5. Complex Problem Solving

Multi-step Algorithms

1. Breaking complex problems into smaller components
2. State management and data flow design
3. Recursive thinking and iterative solutions

Game Logic and Simulations

1. Grid-based problems and coordinate systems
2. Rule-based systems and logic engines
3. Simulation algorithms and modeling

Data Transformation Projects

1. Format conversion and data mapping
2. Aggregation and reporting logic
3. ETL (Extract, Transform, Load) patterns in ABAP

6. Introduction to SAP S/4HANA

1. What is S/4HANA and its evolution from SAP ERP

2. How ABAP is used in S/4HANA vs traditional environments
3. Key modern paradigms: ABAP Cloud, S/4HANA-ready development, RAP as primary model
4. Practical implications: performance, CDS layers, Fiori integration (overview-level)

Overview of the format

- **Day 1**: Build programming fundamentals—syntax mastery, data types, basic control flow, and simple algorithms.
- **Day 2**: Advance to complex data structures—internal tables mastery, advanced algorithms, and performance optimization.
- **Day 3**: Master problem-solving skills—complex algorithms, multi-step solutions, and professional coding practices, plus S/4HANA context.

© 2026 Faysal Aziz · Kuala Lumpur