

# SAP Back-End Developer - ABAP Cloud Training

---

## Course Description

This course provides a comprehensive introduction to **ABAP in Cloud** and **ABAP programming on SAP Business Technology Platform (BTP)**. This comprehensive course introduces the SAP HANA environment and ABAP (Advanced Business Application Programming) development paradigm, focusing on best practices and tools essential for ABAP programming in the context of SAP HANA. Participants will explore the ABAP programming language, data modeling in ABAP, handling complex internal tables, and integrating ABAP code with SAP HANA. It equips participants with the skills to develop, enhance, and maintain robust applications within the SAP environment. Covering foundational concepts to advanced techniques, the training emphasizes key aspects of ABAP development, including data modeling, database integration, object-oriented programming, performance optimization, and innovations such as ABAP Core Data Services (CDS) and the ABAP RESTful Application Programming Model (RAP). Participants will also explore how to extend SAP business applications to meet specific business needs using the SAP Extension Suite and implement side-by-side extensions through real-world examples.

## Audience Profile

This course is designed for:

- SAP developers seeking to deepen their understanding of ABAP in the cloud.
- Application consultants aiming to enhance their technical expertise.
- System analysts and software engineers transitioning to SAP development on SAP BTP.
- IT professionals interested in ABAP-based application development and integration.

## Prerequisites

Participants should have:

- Basic knowledge of programming concepts.
- Familiarity with database structures and principles.

## Course Objectives

After completing this course, developers, consultants, and architects will be able to:

- Understand the basic components of SAP Business Technology Platform (BTP) and the ABAP environment in the cloud.
- Explain the evolution of ABAP programming and explore the ABAP RESTful Programming Model (RAP).
- Extend SAP business applications to meet specific business needs using the SAP Extension Suite.

- Identify and utilize different extension types such as key user and side-by-side extensibility, and implement side-by-side extensions through an example scenario.
  - Create an ABAP Cloud project in ADT and implement conditional logic in a program to execute actions based on conditions.
  - Work with data types and variables to write basic ABAP code, retrieve data from the database, and use it in programs.
  - Analyze runtime errors, debug an ABAP program, and improve program performance.
- 

## **Table of Contents (TOC)**

### **Module 1: Introduction to ABAP in the Cloud**

- Preparing the Development Environment
- Taking a First Look at ABAP
- Understanding Software Structure and Logistics
- Developing Your First ABAP Program
- Labs:
  - Create an ABAP Cloud project
  - Work with a development object
  - Create an ABAP package
  - Create a "Hello World" application

### **Module 2: Exploring Data Modeling in ABAP**

- Exploring ABAP Dictionary
- Labs:
  - Describe the basic tasks of the ABAP Dictionary
  - Name the most important Dictionary objects

### **Module 3: Creating Database Tables**

- Defining Domains and Data Elements
- Creating Database Tables
- Handling Changes to Database Tables
- Labs:
  - Create domains

- Create data elements
- Create a table definition
- Edit the field list of a database table
- Maintain additional table properties
- Define and use include structures
- Adjust database table changes (direct/indirect)

#### **Module 4: Defining Global Data Types**

- Using Dictionary Objects as Data Types
- Defining Dictionary Structures
- Defining Dictionary Table Types
- Labs:
  - Describe the use of dictionary objects as data types
  - Define dictionary structures
  - Define a dictionary table type
  - Define a nested table type

#### **Module 5: Applying Basic Techniques and Concepts**

- Understanding the Basics of ABAP
- Working With Basic Data Objects and Data Types
- Processing Data
- Using Control Structures in ABAP
- Working with Simple Internal Tables
- Debugging an ABAP Program
- Labs:
  - Describe the evolution of ABAP
  - Declare data objects
  - Assign values
  - Perform arithmetic calculations
  - Apply string processing
  - Implement conditional branching

- Handle Exceptions
- Define simple internal tables
- Enter debugging mode
- Analyze data objects

## **Module 6: Reading Data from the Database**

- Investigating a Table Definition
- Implementing Basic SELECT Statements
- Labs:
  - Investigate a table definition
  - Describe basic features of ABAP SQL
  - Read single values from the database

## **Module 7: Working with Structured Data Objects**

- Declaring a Structured Data Object
- Working with Structured Data Objects
- Labs:
  - Declare a structured data object
  - Work with structured data objects
  - Use structured data objects in ABAP SQL

## **Module 8: Improving Internal Table Performance**

- Processing the Contents of Internal Tables
- Using Field Symbols to Process Internal Tables
- Working with Sorted and Hashed Tables
- Improving Internal Table Performance Using Secondary Keys
- Labs:
  - Process the contents of an internal table
  - Process internal tables using field symbols
  - Work with sorted and hashed tables
  - Improve internal table performance using secondary keys

## **Module 9: Working with Complex Internal Tables**

- Declaring a Complex Internal Table
- Working with Complex Internal Tables
- Labs:
  - Declare a complex internal table
  - Fill complex internal tables with data
  - Access the content of complex internal tables
  - Use complex internal tables in ABAP SQL

### **Module 10: Using Data Types and Type Conversions Correctly**

- Using Data Types and Type Conversions Correctly
- Labs:
  - Classify technical data types in ABAP
  - Avoid the pitfalls of type conversions
  - Calculate with dates, times, and timestamps

### **Module 11: Processing Character Fields**

- Using Translatable Texts and String Functions
- Labs:
  - Use string functions
  - Implement regular expressions

### **Module 12: Working with Local Classes**

- Defining Classes and Methods
- Using Encapsulation
- Labs:
  - Define a local class
  - Call methods in encapsulated classes

### **Module 13: Designing Effective Object-Oriented Code**

- Implementing Inheritance and Interfaces
- Using Factory Methods
- Labs:
  - Implement classes with inheritance

- Use factory methods

#### **Module 14: Defining and Working with Exception Classes**

- Defining and Using Exception Classes
- Labs:
  - Create custom exception classes
  - Implement exception handling techniques

#### **Module 15: ABAP Core Data Services**

- Exploring Core Data Services and Objects
- Labs:
  - Utilize CDS objects

#### **Module 16: Defining Basic CDS Views**

- Defining and Using CDS Views
- Applying Annotations
- Building Views on Views
- Labs:
  - Create and analyze CDS views
  - Use annotations

#### **Module 17: Defining Relationships and Associations**

- Creating Relationships and Associations in CDS
- Labs:
  - Define relationships and associations
  - Use path expressions

#### **Module 18: Using Code Pushdown in CDS Views**

- Implementing SQL Expressions and Built-in Functions
- Defining CDS Views with Input Parameters
- Labs:
  - Use SQL expressions, aggregations, and input parameters

#### **Module 19: Defining Meta Objects for Dictionary Objects and CDS Views**

- Implementing Access Controls and Extending CDS Views

- Labs:
  - Extend dictionary objects and CDS views

#### **Module 20: Analyzing and Testing Code**

- Using ABAP Test Cockpit and ABAP Unit
- Profiling and SQL Trace
- Labs:
  - Conduct code analysis and profiling

#### **Module 21: Using Code Pushdown in ABAP SQL**

- Implementing Joins, Expressions, and Aggregations
- Labs:
  - Perform calculations and retrieve optimized data

#### **Module 22: Implementing Authorization Checks**

- Describing Authorization Concepts
- Using AUTHORITY-CHECK
- Labs:
  - Implement CDS access controls and authorization checks

#### **Module 23: Adding Documentation to ABAP Code**

- Documenting ABAP Code
- Labs:
  - Create comprehensive documentation

#### **Module 24: Implementing Database Updates Using Business Objects**

- Analyzing and Updating Data Using EML
- Labs:
  - Use EML statements

#### **Module 25: Describing the ABAP RESTful Application Programming Model**

- Exploring the RAP Architecture
- Enhancing User Experience
- Labs:
  - Develop OData services

- Implement business object behavior