Intermediate ABAP Programming

Course Description:

The **Intermediate ABAP Programming** course is designed to build upon your foundational knowledge of ABAP and equip you with advanced skills required for efficient and effective programming within the SAP environment. This course delves into more complex aspects of ABAP, including code analysis, optimization techniques, advanced data handling, object-oriented programming, and more. By the end of this course, you will be capable of writing robust, optimized, and maintainable ABAP code, which is essential for complex SAP applications.

Audience Profile:

This course is ideal for:

- ABAP developers who have completed basic ABAP training and are looking to enhance their skills.
- SAP professionals involved in the development, testing, or maintenance of ABAP code.
- Technical consultants who require deeper knowledge of ABAP programming for custom SAP solutions.

Prerequisites:

Participants should have:

- Basic understanding of ABAP programming.
- Familiarity with SAP systems and basic data handling within SAP.
- Completion of a beginner-level ABAP course or equivalent experience.

Course Objectives:

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

1. Create and manage ATC check variants and perform static code checks.
2. Implement and execute ABAP unit tests and profile ABAP programs.
3. Detect and analyze performance bottlenecks using ABAP Profiling and SQL Trace.
4. Classify and manage ABAP data types and perform accurate type conversions.
5. Process character fields, strings, and implement translations within ABAP developments.

6. Optimize ABAP SQL code with code pushdown techniques and advanced SQL functionalities.
7. Enhance internal table performance with advanced table processing techniques.
8. Implement effective authorization checks and manage user access controls within ABAP.
9. Design and implement object-oriented ABAP code, including inheritance, interfaces, and factory methods.
10. Define and handle exception classes and incorporate robust error-handling mechanisms in ABAP.
11. Document ABAP code efficiently for better maintainability and knowledge transfer.

Table of Contents:

Labs:

- Classify technical data types in ABAP
- Avoid the pitfalls of type conversions
- Calculate with dates, times, and timestamps

Unit 3: Processing Character Fields

- Using Translatable Text in ABAP
- Processing Strings Using Functions and Regular Expressions

Labs:

- Describe the translation process for ABAP developments
- Use text elements for making developments translatable
- Describe built-in string functions in ABAP
- Work with built-in string functions in ABAP
- Explain the use of regular expressions in ABAP

Unit 4: Using Code Pushdown in ABAP SQL

- Implementing Joins
- Working with Expressions in ABAP SQL
- Performing Calculations and String Processing in ABAP SQL
- Using Special Built-in Functions in ABAP SQL
- Sorting and Condensing Data Sets in ABAP SQL

Labs:

- Implement joins
- Differentiate between inner joins and outer joins
- Implement nested joins
- Use some simple expressions in ABAP SQL
- Perform calculations on the database
- Perform string processing on the database
- Process dates, times, and timestamps on the database

- Use built-in conversion functions
- Request sorted result sets from the database
- Retrieve condensed and aggregated data sets

Unit 5: 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

Unit 6: Implementing Authorization Checks

- Describing the Authorization Concept in ABAP
- Using CDS Access Controls
- Using the AUTHORITY-CHECK Statement

Labs:

- Describe the authorization concept in ABAP
- Use CDS access controls
- Use the AUTHORITY-CHECK statement

Unit 7: Designing Effective Object-Oriented Code

- Implementing Inheritance
- Using Inheritance
- Defining Interfaces
- Using Interfaces

- Implementing Factory Methods

Labs:

- Implement a specialized class
- Use inheritance
- Define interfaces
- Use interfaces
- Use factory methods

Unit 8: Defining and Working with Exception Classes

- Working with Exception Classes
- Defining Your Own Exception Classes

Labs:

- Work with exception classes
- Define your own exception classes

Unit 9: Adding Documentation to ABAP Code

- Documenting ABAP Code

Labs:

- Document ABAP code