# SAP ABAP on HANA — Course Syllabus

## Module 1 — Basics of SAP & ABAP

- Introduction to SAP — System, Applications and Products
- Introduction to ABAP (Advanced Business Application Programming)
- SAP system architecture & landscape
- Application server architecture
- Types of SAP GUI; installation & logon (SAP Logon Pad)
- SAP GUI features and elements; theme settings
- SAP Easy Access overview
- ABAP Workbench and common transaction codes
- Session management: logon, session details, logoff steps
- F1 / F4 help in SAP
- SAP-specific objects vs custom objects; introduction to RICEF/W
- SAP Object Navigator, Packages and Transport Requests
- Header-item table concepts and types of SAP projects

## Module 2 — ABAP Dictionary & Data Modeling

- Domains and Data Elements — creation and usage
- Table creation: header and item tables; primary & foreign keys
- Delivery class, data browser / table view options
- Types of SAP database tables and technical settings (data class, size category, DB properties)
- Enhancement category; client (MANDT) field
- Structures, INCLUDE and APPEND
- Views: Database View, Projection View, Maintenance View, Help View — when and how to use
- Indexes; buffering concepts
- Search help and types
- Lock objects: introduction, modes, creation, and use in programs
- Table Maintenance Generator (TMG) and its events
- Text tables and database utility functions

## Module 3 — ABAP Programming Fundamentals

- ABAP Editor introduction; program creation

- Data types and data objects; system variables
- Control flow: IF, CASE, DO, WHILE; loop control: CONTINUE, CHECK, EXIT
- String operations: CONCATENATE, SPLIT, FIND, SHIFT, SUBSTRING, REPLACE
- String comparison operators (CO, CN, CA, NA, CS, NS, CP, NP)
- Internal tables & work areas — concepts and patterns
- Internal table operations: SELECT, READ, MODIFY, DESCRIBE
- Types: Standard, Sorted, Hashed; header line vs without header line
- Comparison of table types
- Selection-screen programming: PARAMETERS, SELECT-OPTIONS, blocks, frames, titles, text symbols
- Field symbols: ASSIGN, use in LOOP/READ/MODIFY/APPEND

## Module 4 — Reporting (Classical, Interactive, ALV)

### Classical & Interactive Reports

- Classical report basics: single and multiple DB table reports
- Joins: inner, left outer; column labels, SY-ULINE, SY-VLINE
- Message classes; transaction code creation for executable programs
- Report events: LOAD, INITIALIZATION, AT SELECTION-SCREEN, START-OF-SELECTION, END-OF-SELECTION, TOP-OF-PAGE, END-OF-PAGE, etc.
- Selection-screen events: value/help requests, on field handling
- Interactive report events: AT PF, AT LINE-SELECTION, HIDE vs GET CURSOR, AT USER-COMMAND, TOP-OF-PAGE during line selection
- Control break statements: AT FIRST, AT LAST, AT NEW <field>, AT END OF <field>
- Variants for reports; common performance guidelines; page overflow analysis

### ABAP List Viewer (ALV)

- ALV introduction and features (list, grid, hierarchical, block)
- Creating ALV: REUSE_ALV_LIST_DISPLAY, REUSE_ALV_GRID_DISPLAY, hierarchical ALV (REUSE_ALV_HIERSEQ_LIST_DISPLAY)
- Field catalog: LVC_FIELDCATALOG_MERGE and REUSE_ALV_FIELDCATALOG_MERGE; modifying field catalog
- ALV functionality: sorting, filtering, top-of-page, interactive events, hotspot, double-clicks
- PF status, user commands and copying standard PF status

- ALV by classes: CL_GUI_ALV_GRID and CL_SALV_TABLE (factory method): column settings, sorting, filtering, events
- ALV bug analysis and common solutions

## Module 5 — Debugging & Tools

- Debugging fundamentals; executable keys (F5/F6/F7/F8)
- Static and dynamic breakpoints; setting/deleting breakpoints (keyword, function module, message)
- Watchpoints; SAP memory vs ABAP memory checking
- Session vs external breakpoints; desktop management (DESKTOP1/2/3)
- Debugging dialogs/pop-ups and update debugging (stop on selection/updation of tables)
- Ways to skip authority-check; editing DB records (insert, edit, delete) during debugging

## Module 6 — Modularization & Reuse

- INCLUDE programs and subroutines (CALL/FORM); pass by value vs reference
- Function modules: creating, testing, saving parameters as test data
- Comparison: INCLUDE vs Function Module vs Subroutine
- Function groups and best practices

## Module 7 — File Handling & Archiving

- File handling on presentation and application servers
- File upload/download; file archiving basics

## Module 8 — Data Migration & Interfaces

- Data migration techniques overview
- Batch Data Communication (BDC): recording, CALL TRANSACTION vs SESSION method, file upload, recording-to-code changes
- Legacy System Migration Workbench (LSMW) basics and batch input recording
- BAPI: creation, extension (example: material creation), custom fields via BAPI/SPRO
- BAPI vs RFC comparison; RFC creation and usage
- ALE/EDI, IDoc architecture and processing; IDoc ports, partner profiles, process codes

- IDoc: generating interfaces from BAPI, WE19 testing, BD87 error reprocessing, reprocessing with BDCP2/BD61/BD50/BD52 change pointers
- Data migration using Business Object of LSMW

## Module 9 — Enhancements & Modifications & Exits

- Enhancements: implicit and explicit enhancement points
- Customer exits: function module exit, screen exit — finding and implementing exits
- BAdI: classical vs kernel (new) BAdIs; comparison and implementation details
- Customized new BAdI: definition, implementation, filters, fallback class
- User-exits: finding and using exits
- Business Transaction Events (BTE): types, finding P/S interface and process interface, implementation and comparison with BAdI

## Module 10 — Forms (SmartForms & Adobe Forms)

### SmartForms

- SmartForms fundamentals: creation, windows, text types (text element, text module, include text)
- Calling SmartForms from driver programs; selection logic and currency/quantity fields
- Debugging SmartForms; converting output to PDF; dynamic graphics; barcodes
- SmartStyles, output options, language translation of text modules, sending as email attachments

### Adobe Forms

- Adobe Forms introduction; interface & layout design (master page)
- Executing Adobe Forms; layout elements, scripting languages
- Integration into programs; barcodes; text includes and translations
- Sending Adobe forms as email attachments; dynamic graphics

## Module 11 — Transport Organizer & Version Management

- Transport Organizer basics: Workbench requests, transport request statuses (modifiable, released)
- Version management; customizing request; transport of copies
- Moving objects between TRs; changing TR status (released ⇄ modifiable); moving local objects into packages

## Module 12 — Performance Tools & Optimization

- ST05 (SQL trace); SAT (runtime analysis); EPC (extended program check)
- Code performance guidelines; ABAP profiling best practices

## Module 13 — Cross-Application Technologies

- Cross-application concepts; BAPI creation of business objects/functions/tables
- IDoc / BAPI integration details (revisit): ALE/EDI, message types, segments, BDBG generation
- RFC, ALE client-to-client data transfer using RBDMIDOC; scheduling background jobs (RBDMIDOC)
- Change pointers and change document logs; configuration steps for BD61/BD50/BD52

## Module 14 — Introduction to HANA & ABAP on HANA

- ABAP on HANA overview: row store vs column store, data compression, OLTP vs OLAP
- In-memory DB concepts; table partitioning, insert-only approaches
- Code pushdown principles; table enhancement in SE11 for HANA
- SQL performance rules; classical Open SQL limitations vs native SQL
- Open SQL enhancements: literals, arithmetic/aggregation/string expressions, CASE, COALESCE, host expressions in WHERE, client handling
- Join improvements: CROSS, RIGHT OUTER JOIN; GROUP BY, HAVING, UNION/UNION ALL
- Window expressions in ABAP SQL; CTEs (Common Table Expressions)

## Module 15 — Modern ABAP (7.4 / 7.5) & New Syntax

- New syntax features: inline declarations, COND, SWITCH, significance of #
- String expressions, VALUE and BASE in value expressions, corresponding operator
- FOR expressions, ALPHA keyword replacement, NEW keyword with parameters, LET, CONV operator
- REDUCE and FILTER operators; table expressions (LINE_EXISTS) and TRY/CATCH usage with VALUE operator

- Advanced Open SQL: conditional expressions, aggregate functions, element list after FROM clause, comma-separated columns, join improvements and * usage
- Writing reports using new syntax

## Module 16 — CDS Views & Access Control

- Introduction to Core Data Services (CDS) and DDL annotations (DDIC-based)
- CDS entities vs DDIC-based CDS; object model and ignore-propagated annotations
- Joins, literals, input parameters, cardinality, associations and path expressions
- Using CDS in SELECT queries; element annotation, unit & currency conversions
- CDS table functions and when to use them; client handling in table functions
- CDS access control using DCL (PFCG condition, inherit condition, user/aspect/literal conditions)
- ABAP CDS vs Open SQL; exposing CDS as OData and mastering OData using RDS

## Module 17 — ABAP Managed Database Procedures (AMDP)

- AMDP introduction and options; creating and calling AMDPs
- Calling CDS from AMDPs and calling AMDPs inside other AMDPs
- Handling select-options in AMDPs; scalar and tabular AMDP functions
- CDS table functions and AMDP framework interactions

## Module 18 — OData & SAP Gateway

- OData fundamentals (a RESTful API): HTTP request/response, service and metadata document
- OData deployment options and creating first OData service
- MPC, DPC, DPC_EXT classes: read and CRUD operations
- Associations, navigation, $expand, deep insert and media handling
- Query options: $select, $count, $links, $value, $batch ($batch performance), $orderby, filtering, skip/top/inlinecount, skiptoken
- Mastering function imports, media handling, concurrency (ETag) and deep inserts
- SmartForm/Adobe Forms handling via OData

## Appendix — Recommended Hands-on Exercises

- Build a classical report with selection screen and variants
- Create a table + search help + table maintenance generator
- Build an ALV grid with custom field catalog and event handling
- Implement a SmartForm and an Adobe Form and convert to PDF
- Create a CDS view with annotations and expose as OData service
- Implement an AMDP function and compare native SQL performance
- Migrate sample data using BDC and LSMW; create a BAPI for material creation
- Implement a BAdI and a custom enhancement point