

AI-Assisted ABAP/RAP/CAP

COURSE CONTENT

GET IN TOUCH



Multisoft Systems
B - 125, Sector - 2, Noida



(+91) 9810-306-956



info@multosoftsystems.com



www.multosoftsystems.com

About Multisoft

Train yourself with the best and develop valuable in-demand skills with Multisoft Systems. A leading certification training provider, Multisoft collaborates with leading technologies to bring world-class one-on-one and certification training. Intending to empower professionals and businesses across the globe, we offer more than 1500 training courses, which are delivered by Multisoft's global subject matter experts. We offer tailored corporate training, a comprehensive learning solution with lifetime e-learning access, after-training support and globally recognized training certificates.

About Course

AI-Assisted ABAP/RAP/CAP is a modern SAP development course focused on combining established SAP programming skills with emerging AI-assisted software engineering practices. It covers the progression from modern ABAP and ABAP Cloud to RAP-based transactional applications and CAP-based cloud services and extensions.

The course emphasizes practical use of AI throughout development—from understanding requirements and generating initial code to debugging, testing, optimization, and documentation. It also highlights the importance of human review, secure development, clean-core principles, and architectural decision-making, enabling participants to use AI as an effective assistant while retaining control over the quality and reliability of enterprise SAP solutions.

Module 1: SAP Development Landscape and AI-Assisted Development

- ✓ Evolution of SAP application development
- ✓ Traditional ABAP vs. modern SAP development approaches
- ✓ Introduction to ABAP Cloud, RAP, and CAP
- ✓ SAP S/4HANA and SAP BTP development landscape
- ✓ Clean-core principles and extensibility approaches
- ✓ Role of AI in the software development lifecycle
- ✓ Generative AI use cases for SAP developers
- ✓ AI-assisted coding, explanation, testing, and documentation
- ✓ Developer responsibilities when using generated code

Module 2: Modern ABAP Programming Fundamentals

- ✓ Modern ABAP syntax and programming concepts
- ✓ Data declarations and inline declarations
- ✓ Internal tables and structures
- ✓ Expressions and constructor operators
- ✓ String and data processing
- ✓ Open SQL and modern SQL syntax
- ✓ Exception handling
- ✓ Reusable programming structures
- ✓ Performance-aware ABAP programming

Module 3: Advanced ABAP and ABAP Objects

- ✓ Object-oriented ABAP concepts
- ✓ Classes, objects, methods, and interfaces
- ✓ Inheritance and polymorphism
- ✓ Visibility and encapsulation
- ✓ Exception classes
- ✓ ABAP interfaces
- ✓ Reusable object-oriented components
- ✓ Design considerations for enterprise applications
- ✓ Unit-testable ABAP design

Module 4: ABAP Development Tools and Eclipse

- ✓ Introduction to ABAP Development Tools (ADT)
- ✓ Eclipse-based ABAP development
- ✓ Creating and managing ABAP development objects
- ✓ Repository navigation
- ✓ Source code editing and syntax checks
- ✓ Search and navigation capabilities
- ✓ Debugging using ADT
- ✓ Code analysis and quality checks
- ✓ Working with transportable development objects

Module 5: ABAP Cloud and Clean-Core Development

- ✓ ABAP Cloud development model

- ✓ Cloud-ready ABAP principles
- ✓ Released APIs and development objects
- ✓ Restricted ABAP language concepts
- ✓ API-based development
- ✓ Extensibility models
- ✓ Developer extensibility
- ✓ Side-by-side vs. on-stack extensibility
- ✓ Clean-core considerations
- ✓ Designing upgrade-stable extensions
- ✓ Migration considerations for custom developments

Module 6: AI-Assisted ABAP Development

- ✓ AI-assisted ABAP development workflow
- ✓ Generating ABAP code from functional requirements
- ✓ Code completion and coding suggestions
- ✓ Explaining existing and legacy ABAP code
- ✓ Refactoring ABAP using AI assistance
- ✓ Identifying potential code-quality issues
- ✓ Supporting debugging and root-cause analysis
- ✓ Generating unit-test scenarios
- ✓ Creating technical documentation
- ✓ Prompt patterns for SAP development tasks
- ✓ Reviewing and validating AI-generated ABAP
- ✓ Security and governance considerations

Module 7: SAP RESTful Application Programming Model (RAP)

- ✓ Introduction to SAP RAP
- ✓ RAP architecture and development flow
- ✓ RAP programming model components
- ✓ Managed and unmanaged implementation scenarios
- ✓ RAP development artifacts
- ✓ Transactional application development
- ✓ RAP within ABAP Cloud
- ✓ RAP application architecture
- ✓ Understanding end-to-end RAP development

Module 8: CDS and RAP Business Object Modeling

- ✓ Core Data Services fundamentals
- ✓ CDS entities and data models
- ✓ Associations and compositions
- ✓ CDS annotations
- ✓ View entities
- ✓ Data modeling for transactional applications
- ✓ RAP business object concepts
- ✓ Root and child entities
- ✓ Projection views
- ✓ Metadata extensions
- ✓ Access control concepts
- ✓ Designing reusable semantic data models

Module 9: RAP Behavior and Service Development

- ✓ Behavior definitions
- ✓ Behavior implementations
- ✓ Managed business objects
- ✓ Unmanaged business objects
- ✓ CRUD operations
- ✓ Determinations
- ✓ Validations
- ✓ Actions
- ✓ Feature control
- ✓ Numbering concepts
- ✓ Draft-enabled applications
- ✓ Locking and concurrency concepts
- ✓ Service definitions
- ✓ Service bindings
- ✓ OData service exposure
- ✓ Testing RAP services

Module 10: AI-Assisted RAP Application Development

- ✓ Converting business requirements into RAP artifacts
- ✓ AI assistance for CDS modeling
- ✓ Generating CDS structures and annotations
- ✓ Supporting behavior definition creation
- ✓ AI-assisted implementation logic
- ✓ Generating validations and determinations

- ✓ Assistance with service definitions
- ✓ Explaining RAP errors and runtime issues
- ✓ AI-supported debugging
- ✓ Refactoring RAP implementations
- ✓ Test-case generation
- ✓ Documentation assistance
- ✓ Validating generated RAP artifacts

Module 11: SAP Cloud Application Programming Model (CAP)

- ✓ Introduction to SAP CAP
- ✓ CAP architecture and development approach
- ✓ CAP within SAP BTP
- ✓ CAP project structure
- ✓ Core Data Services in CAP
- ✓ Entities and relationships
- ✓ Domain models
- ✓ Service models
- ✓ Node.js and Java development options
- ✓ CAP development lifecycle
- ✓ Local development and testing

Module 12: CAP Services, APIs, and Application Logic

- ✓ Defining CAP services

- ✓ Service exposure
- ✓ CRUD processing
- ✓ Custom handlers
- ✓ Events and actions
- ✓ Functions
- ✓ Business logic implementation
- ✓ OData services
- ✓ REST-based integration concepts
- ✓ Consuming external services
- ✓ Error handling
- ✓ Transactions
- ✓ Application configuration
- ✓ Testing CAP services

Module 13: CAP Security, Integration, and SAP BTP Deployment

- ✓ CAP authentication and authorization
- ✓ Roles and access restrictions
- ✓ SAP BTP service integration
- ✓ Destinations and connectivity concepts
- ✓ External API integration
- ✓ SAP S/4HANA integration scenarios
- ✓ Application configuration for deployment
- ✓ Cloud Foundry deployment concepts
- ✓ Multitarget applications

- ✓ Environment configuration
- ✓ Logging and troubleshooting
- ✓ Application lifecycle considerations

Module 14: Generative AI Across ABAP, RAP, and CAP

- ✓ AI-assisted development lifecycle
- ✓ Requirement-to-code workflows
- ✓ Prompt engineering for SAP developers
- ✓ Context-aware development prompts
- ✓ Code generation across ABAP, RAP, and CAP
- ✓ Code explanation and modernization
- ✓ Generating CDS models from requirements
- ✓ API and service scaffolding assistance
- ✓ Automated documentation support
- ✓ Test-data and test-case generation
- ✓ Debugging assistance
- ✓ Code review and optimization
- ✓ Comparing AI-generated implementation approaches
- ✓ Hallucination and incorrect-code risks
- ✓ Data privacy and intellectual-property considerations
- ✓ Human validation and governance
- ✓ Responsible use of generative AI in enterprise development

Module 15: End-to-End Development Scenarios and Capstone Project

- ✓ Translating business requirements into technical design
- ✓ Selecting ABAP, RAP, or CAP for different requirements
- ✓ Designing clean-core-compatible solutions
- ✓ Building the application data model
- ✓ Implementing business logic
- ✓ Creating and exposing services
- ✓ Integrating SAP and external applications
- ✓ Applying AI assistance during development
- ✓ AI-assisted debugging and optimization
- ✓ Testing and code-quality validation
- ✓ Security considerations
- ✓ Deployment considerations
- ✓ Technical documentation
- ✓ End-to-end project review
- ✓ Development best practices and lessons learned