

**Introduction to Object-Oriented Analysis and Design with UML using Java**

**Course Number:** JAV-200  
**Duration:** 4 days

**Overview**

A good understanding of object-oriented analysis and design is important in designing effective systems using modern software engineering languages and frameworks such as C++, .NET and Java™. This five-day course teaches you how to use object-oriented techniques to analyze real-world requirements and to design solutions that are ready to code. The course employs Unified Modeling Language, using UML 2.0 notation.

**Prerequisites**

Some programming experience is required.

**Materials**

All attendees receive comprehensive courseware covering all topics in the course.

**Software Needed on Each Student PC**

* Windows, macOS, or Linux with at least 8GB RAM
* A recent JDK version installed
* The Java tool the students are likely to use after the class (Eclipse or IntelliJ IDEA are recommended, but other tools are supported)
* Microsoft Access, SQL Server, or another relational database accessible via JDBC or ODBC

**Objectives**

* Learn to use the object oriented development process and to generate a use case/responsibility driven design
* Create objects and classes that apply to object-oriented principals for structure and information engineering
* Master basic UML including roles, inheritance, and dependency
* Learn to model the analysis and design phase using UML
* Master design patterns and considerations such as refining analysis deliverables, architecture, performances, and packaging
* Learn how to implement and test designs for object-oriented applications

**Outline**

* Development Process Overview
  + What is a process?
  + Sample process models
  + Waterfall
  + Spiral
  + Incremental
  + Iterative
  + Unified
  + Responsibility Driven Design
  + What makes a good process?
  + High level overview of OO Development Process
* Use Case/Responsibility Driven Design
  + Contract based approach
  + Responsibility identification
  + Responsibility allocation
  + Roles, stereotypes and interfaces
  + Collaborations
* Objects and Classes
  + What is an object?
  + Characteristics & behavior
  + Communication
  + What is a class?
  + Template
  + UML Class Diagram
* Object-Oriented Principles
  + Why OO?
  + Structured Engineering and Information Engineering
  + Encapsulation
  + Inheritance
  + Polymorphism
  + Dynamic Binding
  + Abstraction
* Basic UML
  + Association and Link relationships
  + Roles
  + Inheritance
  + Aggregation
  + Constraints
  + Multiplicity
  + Dependencies
* Requirements Gathering: Use Cases
  + Static modeling & dynamic modeling overview
  + What is analysis?
  + Requirements gathering
  + Problem domain
  + Use Cases
  + Determining actors
  + Narrow potential objects to business objects
  + Narratives, scenarios and conversations
  + Use case formats (casual, fully dressed, etc.)
* Static Modeling
  + Lexical analysis
  + Data dictionary
  + CRC Cards
  + Class Diagram
* Dynamic Modeling
  + Use Case Diagram
  + Relationships between Use Cases
  + Communication Diagram
  + Sequence Diagram
  + State Diagram
  + Activity Diagram
* Design Considerations
  + Object design
  + Refining analysis deliverables
  + System Design
  + Architecture
  + Optimization
  + Quality Metrics
  + Reuse
  + Performance
  + Concurrency
  + Persistence
  + Normalization
  + Packaging
  + Component Diagram
  + Prototypes
* Design Patterns
  + History
  + Benefits
  + Catalog
  + Model View Controller
  + Factory
  + Composite
  + Chain of Responsibilities
  + Intercepting Filter
* Implementation and Testing
  + Implementation
  + Parallel Implementation
  + Unit, Integration and System Testing
  + Black Box and White Box Testing
  + Testing Criteria
  + The Test Plan
* Conclusion