

Course Outline

The Unified Modeling Language (UML) for Developers Course BA32: 1 days Instructor Led

About this course

This course will provide developers with an understanding of the UML diagrams which are used to analyze and capture requirements graphically during the Enterprise Analysis phase of the project. Emphasis will be placed on the proper usage of UML notation to correctly convey the meaning of the data, behavior, and interface requirements.

Audience profile

This course is designed for developers who are interested in learning how to use the UML as a graphical modeling technique for defining, analyzing, and capturing requirements visually.

At course completion

After completing this course, students will be able to:

- Requirements Modeling using UML
- Use Cases and Activity Diagrams
- Capturing Nonfunctional Requirements
- Refining the Use Case diagram with «Includes» and «Extends»
- Capturing the Interface Requirements
- Capturing Data Requirements with a Class Diagram

Course Outline

#1 – Course Introduction

- Administration and Introductions
- What is the UML?
- Why are modeling languages important?
- Overview of UML diagrams
- **Workshop:** Ball Toss Challenge

#2 – The Context Diagram

- Utilize a context diagram to define system scope
- Identify system actors and distinguish between primary and secondary actors
- Use generalization categorize actors
- **Workshop:** Draw a System Context Diagram

#3 – The Use Case Diagram

- Why Use Cases?
- Identifying and properly naming Use Cases
- Use Case associations... when and how to use «include» and «extend»
- How to represent Use Case packages
- Potential pitfalls
- **Workshop:** Draw a System Use Case Diagram

Course Outline

#4 – The Activity Diagram

- How activity diagrams are used to detail a Use Case
- Basic UML Activity Diagramming notation
- Swimlanes, activities, branches, merges, splits and joins
- **Workshop:** Draw an Activity Diagram

#5 – The Sequence Diagram

- Objects, events, sequence, and lifelines
- Synchronous vs. asynchronous behavior
- Sequence diagramming notation
- **Workshop:** Draw a Sequence Diagram

#6 – The Class Diagram

- Why analyze data and behavior?
- Performing data analysis
- How to define objects, attributes, and operations
- Links, associations, and multiplicity
- **Workshop:** Draw a Class Diagram

#7 – Capturing Non-Functional Requirements

- Use cases as a context for requirements
- Common approaches to documenting non-functional requirements
- Categories of non-functional requirements
- **Workshop:** Write Non-Functional Requirements