

UML Essentials

An Introduction for Analysis and Design

The Unified Modeling Language (UML) is the industry standard notation for describing object-oriented systems. It can be used as a medium for describing existing systems as well as for communicating the requirements and structure of new ones. The notation includes a number of diagram types that can be applied from analysis, through design and to system deployment. Although independent of development lifecycle, UML is intended to support iterative and incremental, and architecture-centric development.

The *UML Essentials* course presents analysis and design concepts structured around the notation of the new UML 2.0 standard. It develops the concepts and introduces the notation through lectures, discussion, individual exercises and a team-based case study.

Objectives

- Present and apply the UML notation from object-oriented analysis through to deployment
- Model the static and dynamic aspects of an object-oriented system
- Understand when and how to apply UML features and modelling techniques
- Use patterns to refine analysis, design and implementation

Audience

The course is suitable for software developers familiar with object-oriented principles and practices. Prior programming, design or analysis experience is assumed. Any previous experience with object-oriented programming or notations is advantageous.

Content

Development Concepts Lifecycle · Iterative and incremental development · Requirements · Analysis · Design · Architecture · Patterns and idioms · Refactoring · Object orientation

Modelling Purpose of a model · UML · Static and dynamic models · Logical and physical models

Use Cases Scope and system boundary · Actors and use cases · Use case documentation · Use case relationships · Use case instances · Use case guidelines

Activity Activities and transitions · Structuring control flow · Activity diagrams versus flow charts

Classes and Objects Class diagrams · Object diagrams · Operations and attributes · Visibility · Class specialisation and generalisation · Interfaces and realisation · Parameterised classes

Object Relationships Dependency, association, aggregation and composition · Multiplicity · Navigability · The Composite pattern · Constraints · Qualified associations · Association classes

Interaction Sequence diagrams · Communication diagrams · Selection and iteration · Synchronisation and concurrency · Interaction overview diagrams · Timing diagrams

Delegation Collaborations and patterns · Delegation principles · The Adapter pattern · The Proxy pattern · The Null Object pattern · Object factories and the Factory and Disposal Method patterns · The Manager pattern

State Stateful and stateless objects · State charts · Composite states · Concurrent states · The Objects for States pattern · The Collections for States pattern

Packages Dependency management · Package diagrams · The Layers pattern · The Explicit Interface pattern · The Observer pattern · The Role Decoupling pattern

Components and Deployment Component diagrams · Interfaces · Composite structure diagrams · Deployment diagrams · Component environments · Multi-tier architectures

Development Process The Rational Unified Process · Inception, elaboration, construction and transition · Agile processes · Prototyping · Testing and integration · Applying patterns

Additional Details

Duration 4 days

Setup Projection facilities for a laptop · Whiteboards and/or flip charts

Contact Kevlin Henney · kevin@curbralan.com · Curbralan Limited · +44 117 942 2990