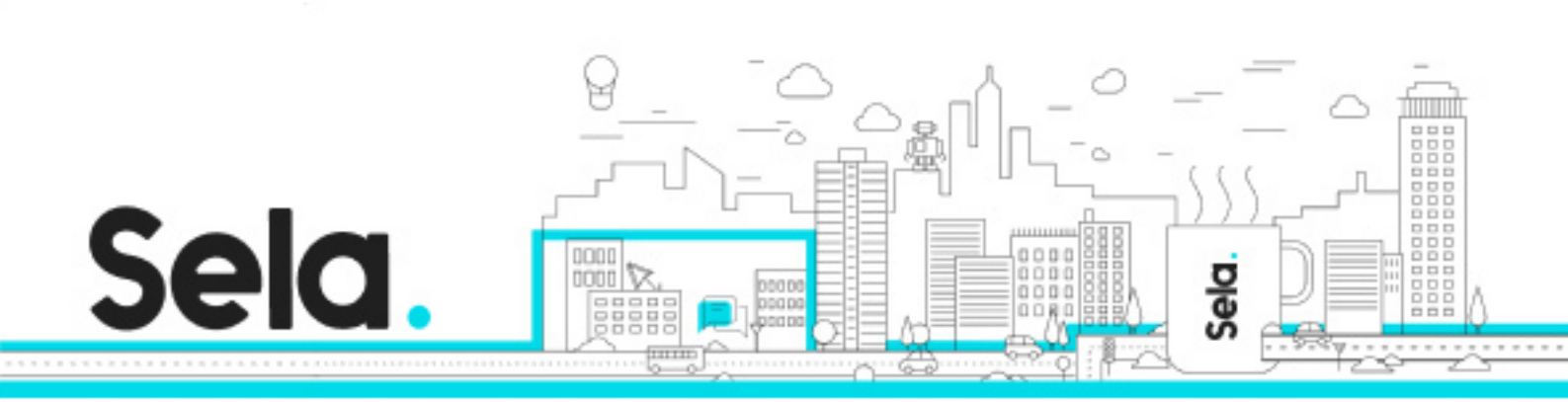
Sela.

UMLR

Object Oriented Analysis / Design using UML

college@sela.co.il 03-6176666



Object Oriented Analysis / Design using UML

UMLR - Version: 2

4 days Course

Description:

This course teaches the evolutionary development approach of Object-Oriented Programming using the Unified Modeling Language, UML version 2.1 The course presents the important topics related to development methodologies using classes and objects, multiple configuration, inheritance, the reusability principle, analysis of requirements using the Use Case Model, realization of Use Case through the process of Object Finding Analysis, and the writing of the solution statistical models, dynamic models, and design patterns. Examples and hands-on exercises are provided for easy assimilation and application of the learning material.

Intended audience:

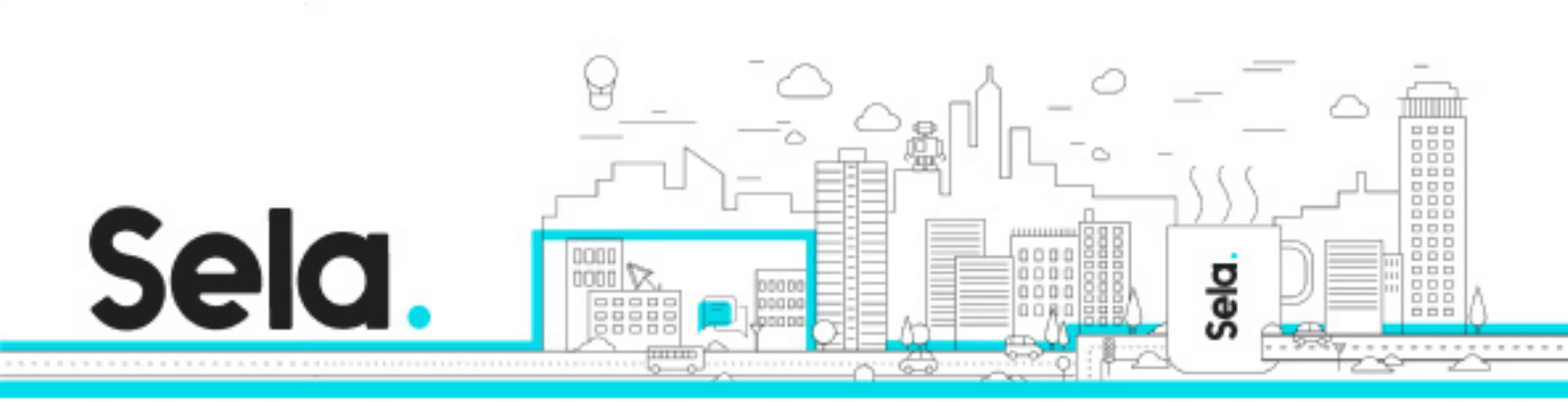
This course is intended for Team Leaders, System Analysts, Project Leaders and Programmers, required to plan requirements and design detailed content using the UML model.

Prerequisites:

Knowledge and experience in an OO programming language.

Objectives:

Topics:



Forward on Object Oriented Technologies and the UML Method.

- Software development process: The Waterfall Model vs. The Spiral Model.
- The Software Crisis, description of the “real world” using the Objects Model.
- Classes, inheritance and multiple configurations.
- Quality software characteristics.
- Description of the Object Oriented Analysis process vs. the Structure Analysis Model.

Introduction to the UML Language.

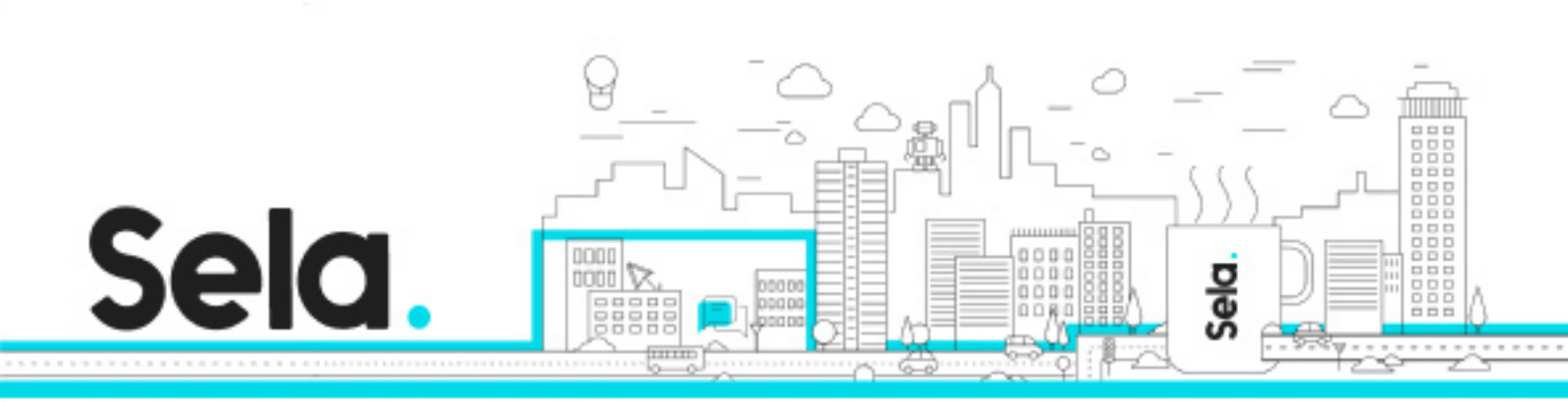
- Standards.
- Elements of the language.
- General description of various models.
- The process of Object Oriented software development.
- RUP process.
- Description of Design Patterns.
- Technological Description of Distributed Systems.

Requirements Analysis Using Case Modeling

- Analysis of system requirements.
- Actor definitions.
- Writing a case goal.
- Use Case Diagrams.
- Use case relationships

Exercise in Case Modeling.

- Actors.



- Use cases.
- Relationships.
- Supplemental document.
- Flow of events.

Transfer from Analysis to Design in the Characterization Stage: Interaction Diagrams.

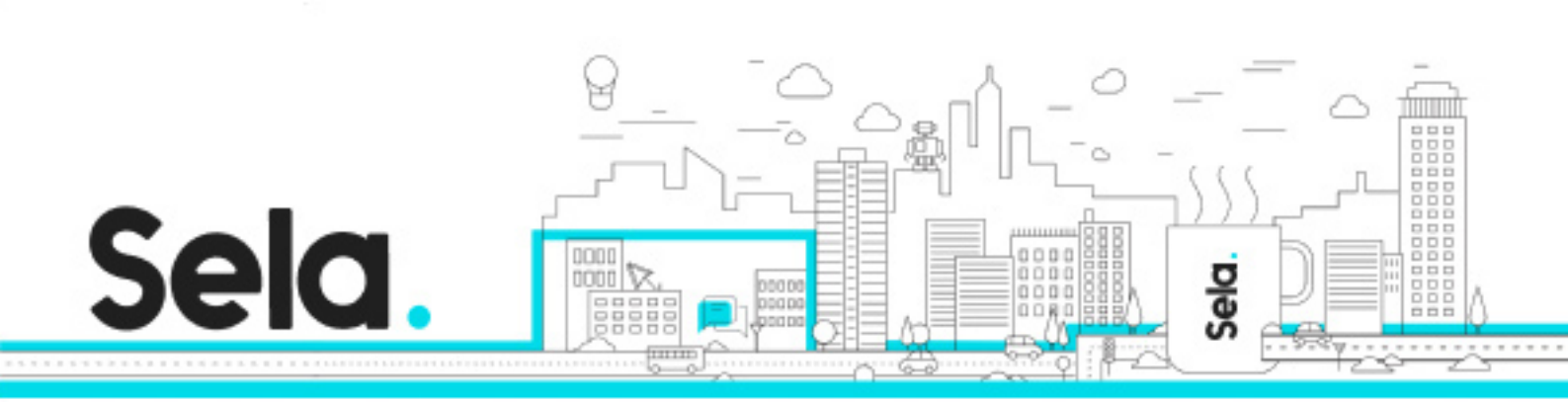
- Description of goal.
- Defining UML Method, Operation, Object Interface, Class.
- Sequence Diagram.
- Finding objects from Flow of Events.
- Describing the process of finding objects using a Sequence Diagram.
- Describing the process of finding objects using a Collaboration Diagram.

The Logical View Design Stage: The Static Structure Diagrams.

- The Class Diagram Model.
- Attributes descriptions.
- Operations descriptions.
- Connections descriptions in the Static Model.
- Association, Generalization, Aggregation, Dependency, Interfacing, Multiplicity.

Exercise in Transfer to Logical View Design in the Characterization Stage.

- Use Case Realization.
- Use Case Realization Package.
- Traceability diagram.
- Creating a Class Diagram Model of the Characterization stage.
- Creating a Class Diagram Model



- Documentation of the Model.
- Creating Connections.
- Creating a VOPC Class Diagram.
- Adding Roll.
- Adding Multiplicity.

Package Diagram Model.

- Description of the model.
- White box, black box.
- Connections between packagers.
- Interfaces.
- Create Package Diagram.
- Drill Down.

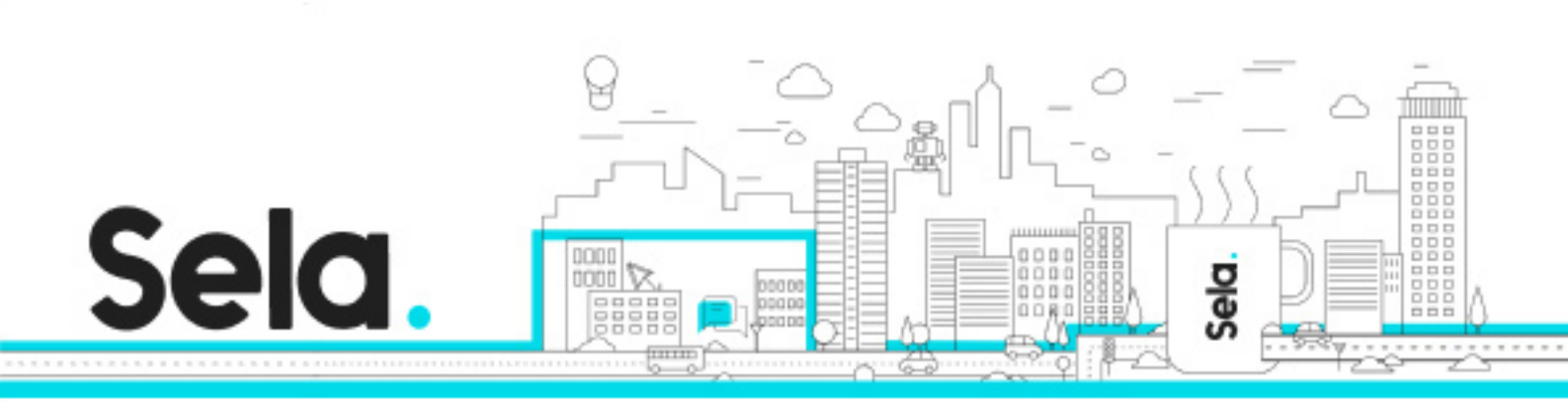
Rose Teamwork.

- Development strategies.
- Dividing the network.
- Control Units.
- Configuration control.

Dynamic Model: State Diagram / Activity Diagram.

- Description of the State Diagram.
- Events Handling.
- Description of the Activity Diagram.
- Exercise in State Machines.

Component Diagram Model.



- Physical Aspect.
- Logical Aspect.
- Connections and Dependencies.
- User face.
- Transfer to the VB, Java and C++ programming language.
- Initial DB design in a UML environment.

Deployment Model.

- Processors.
- Connections.
- Components.
- Tasks.
- Threads.
- Signals and Events.

º Teamwork on the POST Project - Case Study.