



Advanced CASE Technology & Language Systems



Detailed Syllabus for basic **Rhapsody** course

- Introduction
 - Pre-requisites
 - Basic Rhapsody
 - Case study
 - Advanced Rhapsody
 - Specialist Rhapsody
- Part One of Basic Rhapsody
 - Setting up the Rhapsody environment
 - The goal of this section is to understand how to install Rhapsody and how it interacts with the visual studio environment.
- Creating Hello World Project
 - The goal of this section is to master creating classes using the class diagram, generating and modifying the generated code and creating the initial instances via the configuration and component.
- - The browser
 - Class diagram
 - Adding a modifiable constructor
 - Creating a Component
 - Initial instances
 - Selecting an environment
 - Code generation
 - Generated files
 - Editing the generated code
 - Display options on the OMD
 - Displaying the make and main files
 - Project files

- Countdown Project
 - The goal of this section is to add implementation based on the Rhapsody framework and to animate the browser and a statechart.

 - Adding attribute and operations
 - Attribute visibility
 - Operation implementation
 - Active code view
 - A simple Statechart
 - The OXF timer mechanism
 - Generated code
 - Animation
 - Animation tool bar and browser
 - Drawing a sequence diagram
 - Project properties and customization

- Dishwasher Project
 - Dishwasher Statechart
 - Concurrent states
 - History and diagram connectors
 - Adding the transitions
 - Debugging at the design level
 - Creating and running a script
 - Generating events
 - A directed association
 - Code for a relation
 - Adding and accesing event parameters
 - Modifying the main to intialize initial object relations

- Part Two of Basic Rhapsody CashRegister Project
 - The goal of this section is to create a mini project based on the steps of the Rapid Object Oriented Process for Embedded Systems, and to see how the Rhapsody framework implements the various advanced UML artifacts.

 - Use Cases and actors
 - Descriptions
 - Relation between Use Cases
 - Analysis package
 - Navigating the project
 - Adding a public constructor
 - Collections for an association with multiplicity

An ordered and qualified association
Auto generated operations for collections
Composite class
Viewing the objects and relations in the animated browser
Sequence diagrams for the various scenarios
Changing a message to an event
Including iostream.h to the Display class
Adding actions to the transitions
Automatically generated sequence diagrams
Injecting events
Automating the tests
Substatecharts
Active classes
The execute() operation of OMThread
Thread properties
Setting the animation focus on a specific thread
Adding breakpoints
Referenced sequence diagram
Generalization
Directed aggregation
Abstract operations
Optimisation: get() and set()
Generating accessors
Optimisation: avoiding new
Embedded classes

- Part Three Advanced Rhapsody
 - The goal of this section is to gain a greater understanding of the underlying philosophy and implementation of the Rhapsody framework so that we can better tailor it to our specific project and company needs.

The Object eXecution Framework
Event driven framework
OMReactive, OMThread and container classes
Operating system adapter layer
Triggered operations and sending events
Using the STL

- Introduction to Configuration management
 - The goal of this section is to learn how the Rhapsody project division can easily integrate into our general project plan and configuration management scheme.



Advanced CASE Technology & Language Systems



- CM Units
- Command line interface
- Using the SCC interface
- Admin and archive operations
- User defined CM operations using the site.prp file
- DiffMerge tool
- Comparing diagrams
- Working with large projects
- Loading Units
- Useful Tips
- Useful naming conventions