BC401
ABAP Objects

COURSE OUTLINE

Course Version: 15
Course Duration: 5 Day(s)
SAP Copyrights and Trademarks

© 2015 SAP SE. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors.

- Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

- IBM, DB2 Universal Database, System i, System i5, System p, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

- Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

- Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

- Oracle is a registered trademark of Oracle Corporation

- UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

- Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

- HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

- Java is a registered trademark of Sun Microsystems, Inc.

- JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

- SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE in Germany and other countries.

- Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

- Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase, Inc. Sybase is an SAP company.
All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP SE and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.
Typographic Conventions

American English is the standard used in this handbook. The following typographic conventions are also used.

<table>
<thead>
<tr>
<th>Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>This information is displayed in the instructor’s presentation</td>
<td>![Icon]</td>
</tr>
<tr>
<td>Demonstration</td>
<td>![Icon]</td>
</tr>
<tr>
<td>Procedure</td>
<td>![Icon]</td>
</tr>
<tr>
<td>Warning or Caution</td>
<td>![Icon]</td>
</tr>
<tr>
<td>Hint</td>
<td>![Icon]</td>
</tr>
<tr>
<td>Related or Additional Information</td>
<td>![Icon]</td>
</tr>
<tr>
<td>Facilitated Discussion</td>
<td>![Icon]</td>
</tr>
<tr>
<td>User interface control</td>
<td>![Icon]</td>
</tr>
<tr>
<td>Window title</td>
<td>![Icon]</td>
</tr>
</tbody>
</table>

© Copyright. All rights reserved.
## Contents

### Course Overview

<table>
<thead>
<tr>
<th>Unit</th>
<th>Lesson</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction to Object-Oriented Programming</td>
</tr>
<tr>
<td></td>
<td>Lesson: Explaining the Object-Oriented Programming Model</td>
</tr>
<tr>
<td></td>
<td>Lesson: Analyzing and Designing with Unified Modeling Language (UML)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit</th>
<th>Lesson</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Fundamental Object-Oriented Syntax</td>
</tr>
<tr>
<td></td>
<td>Lesson: Creating Local Classes</td>
</tr>
<tr>
<td></td>
<td>Lesson: Creating Objects</td>
</tr>
<tr>
<td></td>
<td>Lesson: Accessing Methods and Attributes</td>
</tr>
<tr>
<td></td>
<td>Lesson: Implementing Constructors in Local Classes</td>
</tr>
<tr>
<td></td>
<td>Lesson: Implementing Class Constructors in Local Classes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit</th>
<th>Lesson</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Inheritance and Casting</td>
</tr>
<tr>
<td></td>
<td>Lesson: Implementing Inheritance</td>
</tr>
<tr>
<td></td>
<td>Lesson: Implementing Upcasts Using Inheritance</td>
</tr>
<tr>
<td></td>
<td>Lesson: Implementing Polymorphism Using Inheritance</td>
</tr>
<tr>
<td></td>
<td>Lesson: Implementing Downcasts Using Inheritance</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit</th>
<th>Lesson</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Interfaces and Casting</td>
</tr>
<tr>
<td></td>
<td>Lesson: Defining and Implementing Local Interfaces</td>
</tr>
<tr>
<td></td>
<td>Lesson: Implementing Polymorphism Using Interfaces</td>
</tr>
<tr>
<td></td>
<td>Lesson: Integrating Class Models Using Interfaces</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit</th>
<th>Lesson</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Object-Oriented Events</td>
</tr>
<tr>
<td></td>
<td>Lesson: Implementing Events in Local Classes</td>
</tr>
<tr>
<td></td>
<td>Lesson: Implementing Events in Local Interfaces</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit</th>
<th>Lesson</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Object-Oriented Repository Objects</td>
</tr>
<tr>
<td></td>
<td>Lesson: Creating Global Classes</td>
</tr>
<tr>
<td></td>
<td>Lesson: Defining and Implementing Global Interfaces</td>
</tr>
<tr>
<td></td>
<td>Lesson: Implementing Inheritance in Global Classes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit</th>
<th>Lesson</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>ABAP Object-Oriented Examples</td>
</tr>
<tr>
<td></td>
<td>Lesson: Using the ABAP List Viewer (ALV)</td>
</tr>
<tr>
<td></td>
<td>Lesson: Describing Business Add-Ins (BAdIs)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unit</th>
<th>Lesson</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Global Classes in ABAP Development Tools</td>
</tr>
<tr>
<td></td>
<td>Lesson: Developing Eclipse-Based ABAP Programs</td>
</tr>
</tbody>
</table>

© Copyright. All rights reserved.
<table>
<thead>
<tr>
<th>Unit</th>
<th>Lesson</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>9:</td>
<td>Class-Based Exceptions</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>Lesson: Explaining Class-Based Exceptions</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>Lesson: Defining and Raising Exceptions</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>Lesson: Implementing Advanced Exception Handling Techniques</td>
</tr>
<tr>
<td>19</td>
<td>10:</td>
<td>Unit Testing</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>Lesson: Unit Testing with ABAP Unit</td>
</tr>
<tr>
<td>21</td>
<td>11:</td>
<td>Object-Oriented Design Patterns</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>Lesson: Implementing Advanced Object-Oriented Techniques</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>Lesson: Implementing the Singleton Pattern</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>Lesson: Implementing Factory Classes Using Friendship</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>Lesson: Implementing Persistent Objects</td>
</tr>
<tr>
<td>23</td>
<td>12:</td>
<td>Runtime Type Services</td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>Lesson: Using Runtime Type Identification (RTTI)</td>
</tr>
<tr>
<td>25</td>
<td>13:</td>
<td>Creation of a Comprehensive Object-Oriented Application</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>Lesson: Developing a Comprehensive Object-Oriented Application</td>
</tr>
</tbody>
</table>
TARGET AUDIENCE
This course is intended for the following audiences:

- Application Consultant
- Development Consultant
- Developer
Lesson 1: Explaining the Object-Oriented Programming Model

Lesson Objectives
After completing this lesson, you will be able to:

- Describe the differences between the procedural and object-oriented programming models

Lesson 2: Analyzing and Designing with Unified Modeling Language (UML)

Lesson Objectives
After completing this lesson, you will be able to:

- Classify objects
- Model in UML
Lesson 1: Creating Local Classes
Lesson Objectives
After completing this lesson, you will be able to:

- Define local classes
- Define attributes
- Create methods

Lesson 2: Creating Objects
Lesson Objectives
After completing this lesson, you will be able to:

- Create objects

Lesson 3: Accessing Methods and Attributes
Lesson Objectives
After completing this lesson, you will be able to:

- Call instance methods
- Call static methods
- Call functional methods
- Access public attributes

Lesson 4: Implementing Constructors in Local Classes
Lesson Objectives
After completing this lesson, you will be able to:

- Create and use constructors

Lesson 5: Implementing Class Constructors in Local Classes
Lesson Objectives
After completing this lesson, you will be able to:

- Create and use static constructors
Lesson 1: Implementing Inheritance
Lesson Objectives
After completing this lesson, you will be able to:

- Explain generalization and specialization
- Implement inheritance
- Access elements of classes in inheritance

Lesson 2: Implementing Upcasts Using Inheritance
Lesson Objectives
After completing this lesson, you will be able to:

- Implement upcasts using inheritance

Lesson 3: Implementing Polymorphism Using Inheritance
Lesson Objectives
After completing this lesson, you will be able to:

- Explain polymorphism
- Implement polymorphism using inheritance

Lesson 4: Implementing Downcasts Using Inheritance
Lesson Objectives
After completing this lesson, you will be able to:

- Implement downcasts using inheritance
- Model class hierarchies
Lesson 1: Defining and Implementing Local Interfaces

Lesson Objectives
After completing this lesson, you will be able to:

- Explain the use of interfaces
- Create generalization and specialization relationships using interfaces

Lesson 2: Implementing Polymorphism Using Interfaces

Lesson Objectives
After completing this lesson, you will be able to:

- Implement polymorphism using interfaces

Lesson 3: Integrating Class Models Using Interfaces

Lesson Objectives
After completing this lesson, you will be able to:

- Implement downcasts with interfaces
- Integrate different submodels using interfaces
- Create and use interface hierarchies
Lesson 1: Implementing Events in Local Classes

Lesson Objectives
After completing this lesson, you will be able to:

- Implement event-controlled method calls
- Trigger and handle events
- Register for events
- Explain visibility sections in event handling

Lesson 2: Implementing Events in Local Interfaces

Lesson Objectives
After completing this lesson, you will be able to:

- Implement events in local interfaces
Lesson 1: Creating Global Classes

Lesson Objectives
After completing this lesson, you will be able to:

• Create global classes
• Test global classes
• Use global classes

Lesson 2: Defining and Implementing Global Interfaces

Lesson Objectives
After completing this lesson, you will be able to:

• Define and implement global interfaces
• Import local classes and interfaces

Lesson 3: Implementing Inheritance in Global Classes

Lesson Objectives
After completing this lesson, you will be able to:

• Generate UML diagrams for global classes
• Implement inheritance in global classes
• Change the display of components in global classes
Lesson 1: Using the ABAP List Viewer (ALV)

Lesson Objectives
After completing this lesson, you will be able to:

- Implement a simple ALV grid
- Handle the double-click event of the ALV grid

Lesson 2: Describing Business Add-Ins (BAdIs)

Lesson Objectives
After completing this lesson, you will be able to:

- Describe BAdIs
Lesson 1: Developing Eclipse-Based ABAP Programs

Lesson Objectives
After completing this lesson, you will be able to:

- Implement Eclipse-based ABAP development
- Use quick fixes and refactoring
Lesson 1: Explaining Class-Based Exceptions

Lesson Objectives
After completing this lesson, you will be able to:

- Explain class-based exceptions
- Handle class-based exceptions
- Debug class-based exceptions

Lesson 2: Defining and Raising Exceptions

Lesson Objectives
After completing this lesson, you will be able to:

- Define global exception classes
- Raise class-based exceptions
- Propagate exceptions

Lesson 3: Implementing Advanced Exception Handling Techniques

Lesson Objectives
After completing this lesson, you will be able to:

- Explain the hierarchy of predefined exception classes
- Explain different ways of handling an exception
- Retry after exceptions
- Implement resumable exceptions
- Map exceptions
Lesson 1: Unit Testing with ABAP Unit

Lesson Objectives
After completing this lesson, you will be able to:

- Perform unit testing
- Use test classes and test methods
- Perform advanced ABAP unit testing
Lesson 1: Implementing Advanced Object-Oriented Techniques

Lesson Objectives
After completing this lesson, you will be able to:

- Implement abstract classes
- Implement final classes
- Access internal tables with object references
- Call navigation methods
- Restrict the visibility of the instance constructor

Lesson 2: Implementing the Singleton Pattern

Lesson Objectives
After completing this lesson, you will be able to:

- Implement factory methods
- Implement the singleton pattern

Lesson 3: Implementing Factory Classes Using Friendship

Lesson Objectives
After completing this lesson, you will be able to:

- Implement friendship relationships

Lesson 4: Implementing Persistent Objects

Lesson Objectives
After completing this lesson, you will be able to:

- Explain persistence services
- Examine persistent classes
- Create persistent objects
- Read data with persistent objects
• Create OO transactions
Lesson 1: Using Runtime Type Identification (RTTI)

Lesson Objectives
After completing this lesson, you will be able to:

- Explain RTTI
- Describe structure type properties at runtime
- Describe object type properties at runtime
Lesson 1: Developing a Comprehensive Object-Oriented Application

Lesson Objectives
After completing this lesson, you will be able to:

• Develop a comprehensive object-oriented application