Microsoft Dynamics

Course Syllabus

Course 80012: Development II in Microsoft Dynamics® AX 2009

Key Data

Course #: **80012**

Number of Days: **2**

Format: **Instructor-Led**

Certification Exams:

This course helps you prepare for the following Microsoft Certified Professional exams:

- AX 2009 Development Introduction

- Development

Certification Track:

This course syllabus should be used to determine whether the course is appropriate for the students, based on their current skills and technical training needs.

Course content, prices, and availability are subject to change without notice.

This 2-day course introduces the student to development in Microsoft Dynamics AX 2009 using X++. This course starts with learning the basics of X++ and its relationship to object-oriented programming in addition to the tools that are required to develop in Microsoft Dynamics AX. The student then learns more about specific control structures, accessing the database using X++, and handling exceptions in Microsoft Dynamics AX. This course is meant to be an introductory course to development in Microsoft Dynamics AX using X++.

Audience

This course is intended for individuals who will be developing within Microsoft Dynamics AX using X++. This audience typically includes technical consultants who will be working with Microsoft Dynamics AX to develop customizations and modifications to meet clients' needs.

This is the second course in the AX Development track and will serve as the entry point for the Development III & IV courses.

Audience

This course will be most beneficial for someone who is new to the concepts of object-oriented programming and programming using X++. Additionally, consultants who are responsible for training or supporting the customer will benefit from this course.

At Course Completion

After completing this course, students will be able to:

- Identify key features of developing with X++
- Describe the basic foundation of object-oriented programming
- Use the development tools available within Microsoft Dynamics® AX
- Create object and data models from existing application elements using the Reverse Engineering tool
- Use best practices to instill good programming habits
- Use the data types that can be used for variables and how to declare and use them
- Use the various operators available and where to use them
- Control program flow using conditional statements in X++
- Call the same blocks of code using Loop statements
- Use standard functions that are built in to the application
- Use output commands to display data and messages to the user
- Use the classes within Microsoft Dynamics AX 2009 X++ development
- Extend a class using the concept of inheritance
- Describe the differences between an object and a class
- Initialize variables in the appropriate place according to scoping rules
- Call methods within the same class
- Use the different method types available
- Describe the similarities and differences between tables and classes
- Retrieve data from the database using a select statement.
- Create, update and delete data in the database.
- Use and build queries using kernel classes.

- Examine the exception handling mechanism in Microsoft Dynamics® AX
- Use the Try, Catch, and Retry commands
- Throw an exception from code.
- Identify and create code used to handle optimistic concurrency exceptions

Prerequisites

Before attending this course, students must have:

- Completed Course 80011: Development I in Microsoft Dynamics AX 2009

In addition, it is recommended, but not required, that students have completed:

- Working knowledge of object oriented programming (OOP)

Student Materials

The student kit includes a comprehensive workbook and other necessary materials for this class.

Module 1: Introduction to X++

This module gives a foundation for understanding development using X++. The student will also be able to use the main development tools in Microsoft Dynamics AX

Lessons
▪ Characteristics X++ ▪ Development Tools ▪ Reverse Engineering ▪ Best Practices
Lab
▪ 1.1 Print to the Screen ▪ 1.2 Debug the Job ▪ 1.3 Create a Data Model ▪ 1.4 Create an XML developer document

After completing this module, students will be able to:

- Identify key features of developing with X++
- Describe the basic foundation of object-oriented programming
- Use the development tools available within Microsoft Dynamics® AX
- Create object and data models from existing application elements using the Reverse Engineering tool
- Use best practices to instill good programming habits

Module 2: X++ Control Statements

This module focuses on programming constructs in X++, such as loops, conditional statements, and functions. In addition, the student will be able to effectively communicate with the end-user using X++ output commands.

Lessons
▪ Introduction to variables ▪ Operators ▪ Conditional statements ▪ Loops ▪ Built-in functions ▪ Communication Tools
Lab
▪ 2.1: Create a Times Table ▪ 2.2: Create a Times Table Using a do while Loop ▪ 2.3: Create a Times Table Using a For Statement ▪ 2.4: Create a YesNo box ▪ 2.5: Create an Infolog Tree ▪ 2.6: Create a dialog box ▪ 2.7: Use X++ Control Statements

After completing this module, students will be able to:

- Use the data types that can be used for variables and how to declare and use them
- Use the various operators available and where to use them
- Control program flow using conditional statements in X++
- Call the same blocks of code using Loop statements
- Use standard functions that are built in to the application
- Use output commands to display data and messages to the user

Module 3: Objects and Classes

This module discusses the concept of object-oriented programming and its relation to X++. By the end of this module, the student will have a solid foundation to develop in Microsoft Dynamics AX using object-oriented concepts.

Lessons
▪ Classes ▪ Inheritance ▪ Objects ▪ Scoping and parameters in X++ ▪ Referencing object methods ▪ Method types ▪ Tables as Classes
Lab
▪ 3.1: Create a new class ▪ 3.2: Instantiate a class ▪ 3.3: Use method parameters ▪ 3.4: Create a run method ▪ 3.5: Create a Calculator Class

After completing this module, students will be able to:

- Use the classes within Microsoft Dynamics AX 2009 X++ development
- Extend a class using the concept of inheritance
- Describe the differences between an object and a class
- Initialize variables in the appropriate place according to scoping rules
- Call methods within the same class
- Use the different method types available
- Describe the similarities and differences between tables and classes

Module 4: Accessing the Database

This module discusses developing modifications that interact with the Microsoft Dynamics AX database. This functionality is frequently needed, making this an important topic in learning development with X++.

Lessons
▪ Retrieving data ▪ Data Manipulation ▪ Queries
▪ **Labs**
▪ 4.1: Retrieving data ▪ 4.2: Update ▪ 4.3: Create a query using X++

After completing this module, students will be able to:

- Retrieve data from the database using a select statement.
- Create, update and delete data in the database.
- Use and build queries using kernel classes.

Module 5: Exception Handling

This module discusses how programs handle exceptions that occur when code is executed. This is an important skill in all programming to make the application execute more efficiently.

Lessons
▪ Exceptions ▪ Try and Catch Commands ▪ Throwing Exceptions ▪ Optimistic Concurrency Exceptions
Lab
▪ **5.1:** Handle an Exception

After completing this module, students will be able to:

- Examine the exception handling mechanism in Microsoft Dynamics® AX
- Use the Try, Catch, and Retry commands
- Throw an exception from code.
- Identify and create code used to handle optimistic concurrency exceptions

Appendix A: Workflow

This Appendix contains an optional self-study lesson on workflow within Microsoft Dynamics AX 2009. By following the procedures in this lesson, students will be able to create and configure simple workflows.

Lessons
▪ Create a workflow category ▪ Create a workflow template ▪ Create a workflow document ▪ Create a workflow approval ▪ Enable workflow on a form ▪ Configure a workflow
Lab
▪ A.1: Add another condition to the Submit Action

After completing this module, students will be able to:

- Identify the components required prior to using workflow
- Specify which applicationmodule a workflow is applicable to using a workflow category
- Create a new workflow template
- Link tables to workflows using a workflow document
- Define what happens when the workflow is approved or denied.
- Apply a workflow to a form
- Configure a workflow