



# 20485: Advanced Windows® Store App Development Using C#

## About this course

This course you will learn advanced programming practices and techniques that will help you develop Windows Store apps. In this course, you will learn how design and develop Windows Store apps, as well as implement advanced features, such as using location information, streaming media to external devices, and integrating with online services. You will also learn how to use Microsoft Visual Studio 2013 and Expression Blend tools.

This course maps to the [70-485](#) exam.

Windows Azure is required or used as part of the lab for this course, which may not be available in all countries. Please check with your Microsoft training provider that this cloud service is available in your area. Training Providers please reference the course setup guide for more details.

## Audience profile

This course is intended for professional developers who have one or more years of experience creating applications for a production environment and who are comfortable programming in C# and XAML to create Windows Store apps.



# 20485: Advanced Windows® Store App Development Using C#

## At course completion

After completing this course, students will be able to:

- Review basic Windows Store app concepts and how Windows Store apps integrate with the Windows 8.1 ecosystem.
- Explore the built-in WinRT animation library and storyboards, and apply transitions and transformations.
- Localize the Windows Store app UI and apply language-specific formatting to a multi-lingual app.
- Create and present a successful splash screen, and differentiate your app with branding.
- Use various storage mechanisms, choose a suitable caching strategy for their app, and use advanced file access methods.
- Work with custom and template controls, and create and consume a WinRT component.
- Print using the Printing API and explore the Play To API.
- Explore the Push Notification Service (WNS), and work with push notifications.
- Use the CameraCaptureUI API and the MediaCapture API.
- Create and consume background tasks.
- Discover sensors, test their availability, and handle sensor events and device availability, and enumerate devices and their capabilities and properties and use the capabilities.



# 20485: Advanced Windows® Store App Development Using C#

## Course Outline

### Module 1: Windows Store App Essentials

In this module, you will get an overview of the Windows 8.1 user experience and features, and will become acquainted with the Windows Store apps and the new user interface. You will review the process of developing apps by using XAML to create the user interface, and the WinRT API to interact with the platform.

#### Lessons

- Review Windows Store App Essentials
- Review Windows Store App Essentials – Under the Hood

After completing this module, students will be able to:

- Describe the Windows 8.1 platform, architecture, and features.
- Describe the process of creating user interfaces with XAML.
- Explain the new WinRT API model.
- Describe the basics of working with files, implementing contracts, displaying notifications, and deploying your app to the Windows Store.

### Module 2: Implementing Animations and Transitions

In this module, you will learn how to add animations to your Windows Store apps, and how to take advantage of the functionality offered by the VisualStateManager class.

#### Lessons

- Using Animations
- Working with Transitions and Transformations

#### Lab : Implementing Animations and Transitions

After completing this module, students will be able to:

- Use both built-in and custom animations, and storyboards.
- Apply transformations to UI elements and use the **VisualStateManager** class to create fluid UI transitions.



# 20485: Advanced Windows® Store App Development Using C#

## Module 3: Implementing Globalization and Localization

In this module you will learn how globalization prepares an app for a global audience. This involves changing culture-specific elements of the user interface, such as dates, times, currencies, languages, and more. Localization takes globalization further by supporting specific languages, cultures, and regions.

### Lessons

- Working with Resource Files
- Implementing Culture-Specific Formatting

Lab : Implementing Globalization and Localization

After completing this module, students will be able to:

- Create and use resource files for localization.
- Implement culture-specific formatting.

## Module 4: Branding and a Seamless UI

This module explains how to brand your app and add a branded splash screen to improve user experience.

### Lessons

- Branding
- Customizing the Splash Screen
- Branding Your App's UI

Lab : Branding and Seamless UI

After completing this module, students will be able to:

- Create a splash screen for a Windows Store app.
- Apply branding to a Windows Store app.

## Module 5: Advanced Data Scenarios in a Windows Store App

In this module, you will learn about the new file storage capabilities of Windows 8.1. In addition, you will learn how to use the file storage capabilities in your apps. You will also learn how to design and implement more useful data storage through caching and how to set up file extensions, associations, and compression. These features will help you to provide a better experience to the user by increasing the responsiveness of your app, and lowering its resource usage.



# 20485: Advanced Windows® Store App Development Using C#

## Lessons

- Windows Store App Storage Options
- Implementing Data Caching
- Advanced File Functionality

### Lab : Caching Data

After completing this module, students will be able to:

- Use file storage in a Windows Store app.
- Explain what caching is, and how and when to use it in a Windows Store app.
- Set up file extensions and associations in an app.
- Compress files in your app.

## Module 6: Creating Reusable Controls and Components

In this module, you will learn how the Windows Runtime (WinRT) contains a rich set of XAML controls, such as GridView, ListView, FlipView, and AppBar. In many cases, you can use these controls to create fully functional apps. However, if the built-in controls do not contain the functionality that you need, you can create your own custom controls. You will be able to reuse these custom controls in other apps, like any other built-in control.

You can also extend the WinRT itself to expose existing or new libraries to Windows Store apps. You can create custom WinRT components, which can be used from any supported programming language (C#, VB.NET, C++, or JavaScript).

### Lessons

- Working with Custom and Templated Controls
- Creating and Using WinRT Components

### Lab : Creating Reusable Controls and Components

After completing this module, students will be able to:

- Create and use custom XAML controls.
- Create and use custom WinRT components.



# 20485: Advanced Windows® Store App Development Using C#

## Module 7: Implementing Advanced Contract Scenarios

This module describes how to use two new contracts:

- The Play To contract enables media sharing with other devices, such as television sets.
- The Print contract enables interaction with printing devices.

Windows Store apps can integrate with the contacts and calendar appointments managed by the operating system. You can create new contacts and calendar appointments, update and delete existing ones.

Lastly, you will learn how to take advantage of several convenient Windows 8.1 application programming interface (APIs) to provide exciting, user friendly features in your apps. Windows RT has native APIs for working with Portable Document Format (PDF) files, creating screenshots of your app, and generating speech from text through speech synthesis.

### Lessons

- The Play To Contract
- The Print Contract
- Integrating Contacts and Calendars
- Support PDF files and screenshots
- Integrating Speech Synthesis

Lab : Integrating the Print Contract Into the App

After completing this module, students will be able to:

- Implement the Play To contract.
- Implement the Print contract.
- Display PDF documents in your app.
- Create snapshots of your app.

## Module 8: The Windows Push Notification Service

This module explains Windows Push Notifications.

### Lessons

- Introduction to The Push Notification Service
- Communicating with the WNS

Lab : The Windows Push Notification Service

After completing this module, students will be able to:



# 20485: Advanced Windows® Store App Development Using C#

- Explain how Windows push notifications work.
- Use Windows push notifications in a Windows Store app.

## Module 9: Capturing Media

In this module, you will learn how to include media capturing functionality in your app. Learning to leverage the Windows Runtime (WinRT) media capturing APIs can help you create an exciting and successful app.

### Lessons

- Using CameraCaptureUI to Capture Pictures, Videos or Audio
- Using MediaCapture to Capture Pictures, Video, or Audio

### Lab : Capturing Media

After completing this module, students will be able to:

- Use the **CameraCaptureUI** API to capture pictures, videos and audio.
- Use the **MediaCapture** API to capture pictures, videos and audio.

## Module 10: Background Tasks

This module explains how to create, register, and use background tasks for Windows Store apps.

### Lessons

- Creating Background Tasks
- Using Background Tasks in a Windows Store App

### Lab : Background Tasks

After completing this module, students will be able to:

- Create background tasks for your Windows Store app.
- Use background tasks in your Windows Store app.

## Module 11: Working with Sensors and Devices

This module describes some of the sensors and devices available to a Windows Store app and explains how to access them from your Windows Store app.

### Lessons



# 20485: Advanced Windows® Store App Development Using C#

- Working with Sensors
- Working with Devices

Lab : Working with Sensors and Devices

After completing this module, students will be able to:

- Discover and access sensors and act upon sensor data from a Windows Store app.
- Discover and interact with devices from a Windows Store app.

## Module 12: Generating Revenue with Your Windows Store App

This module will introduce you to the options you have, as a developer, to make money with your Windows Store apps.

### Lessons

- Implementing Trial Functionality in a Windows Store App
- Implementing In-App Purchases
- Advertising in a Windows Store App

Lab : Generating Revenue with your App

After completing this module, students will be able to:

- Implement trial functionality in your Windows Store app.
- Implement in-app purchases in your Windows Store app.
- Incorporate advertising in your Windows Store app.

## Module 13: Securing Windows Store App Data

In this module, you will learn how to create an application that is secure at the client and server side, and provides protection for your data. You will also learn how to implement a security protocol that lets you communicate with services such as Windows Azure and Microsoft SkyDrive.

### Lessons

- Managing Windows Authentication
- Managing Web Authentication
- Encryption in Windows Store Apps

Lab : Securing Windows Store App Data

After completing this module, students will be able to:



# 20485: Advanced Windows® Store App Development Using C#

- Manage Windows authentication.
- Manage Web authentication.
- Encrypt data.

## Module 14: Tracing and Profiling Windows Store Apps

In this module, you will learn the techniques that help you to analyze your code for potential issues, create run time logs efficiently, and diagnose various types of performance problems in your Windows Store app.

### Lessons

- Analyzing a Windows Store App
- Tracing a Windows Store App
- Profiling a Windows Store App

### Lab : Tracing and Profiling Windows Store Apps

After completing this module, students will be able to:

- Analyze the code of your Windows Store app to find potential problems.
- Trace the run time flow of your Windows Store app.

Diagnose CPU and memory performance problems in your Windows Store app.



# 20485: Advanced Windows® Store App Development Using C#

## Prerequisites

Before attending this course, students must have:

- 6 – 12 months experience programming in C#
- 3 – 6 months experience using Visual Studio 2013 (including experience using preview builds)
- 1 month of experience working on Windows Store apps