

Course Code: CIS160**Course Name: Developing Mobile Apps****Certification: MCSD App Builder – 70-357****Duration: 3 months****Tuition: \$1895****Course Overview**

UWP provides many objects to build layouts that work on a variety of devices. In this course, you will be introduced to UWP, and you will learn about the different types of layouts that can be used to make rich user interface designs. UWP's specialty is being able to run the same app on many different platforms. In this course, you will learn how to use adaptive UI and adaptive coding to make apps that maintain maximum functionality on multiple platforms.

XAML provides many app controls that can be used to implement a variety of features. In this course, you will learn how to use XAML app controls in UWP apps to add functionality.

Navigation between different parts of an app is crucial for development. In this course, you will learn how to use several navigation patterns in a UWP app.

An app's life cycle is important to define functionality, and designing around different events can make features better and more responsive. In this course, you will learn how to program around application life cycle events.

Apps often use various data sources to provide users with rich content and experiences. In this course, you will learn how to use bindings to display various types of data in a UWP app.

User input is crucial to app development in order to allow users to interact with the app. In this course, you will learn how to use traditional input in a UWP app to create interactive features.

On top of traditional methods, devices often provide many other types of input methods which can be used to create rich interactive user experiences. In this course, you will learn how to use a variety of these input methods.

UWP provides easy access to speech APIs and Cortana, for use in creating voice interface features in UWP apps. In this course, you will learn how to use the speech synthesizer, recognize speech, and implement Cortana features.

Allowing apps to communicate with each other and store files makes it easy to expand their experience beyond the local view. In this course, you will learn how to make an app communicate with other apps and how to manage file storage.

Users won't always be in an app to check for updates, so sending them notifications is important to keeping them up to date. In this course, you will learn how to use different types of notifications for UWP.

On top of built-in app functionality, UWP provides features to build background tasks, app services, and reusable components that can interact with an app outside of the app. In this course, you will learn how to use these features.

Security is an important consideration for every app that relies on external data. UWP provides many features for authentication and identity management, which help with security. In this course, you will learn how to use these features.

Course Content

- **Lesson 1 – Introducing UWP and Basic App Layouts**

This lesson covers the following topics:

- describe the UWP platform
- set up Visual Studio to create UWP app projects
- create a simple app and run it on a local machine
- describe UWP responsive UI design
- implement a grid layout in an app
- implement a stack layout in an app
- implement a canvas layout in an app
- implement a relative layout in an app
- manipulate the alignment, margins, and padding of panels to create rich UIs
- implement complex interfaces by combining different types of panels
- lay out a basic app using various types of panels

- **Lesson 2 – Adaptive App Design**

This lesson covers the following topics:

- differentiate between a responsive UI and an adaptive UI
- implement an adaptive UI using the VisualStateManager
- use an AdaptiveTrigger to define states
- set layout properties with AdaptiveComponents
- build tailored layouts for different apps
- use API contracts for adaptive coding
- use type detection for adaptive coding
- use the platform specific analyzer to help write adaptive code
- use different layout techniques to optimize the app's performance
- build an adaptive UI for your apps

- **Lesson 3 – XAML App Controls**

This lesson covers the following topics:

- implement command bars in apps and add buttons using AppBarButton
- use open and closed states and command bar options to handle overflow when the screen is small
- implement flyouts in your app layout
- implement context menus for button events
- implement dialogs using the ContentDialog object

- implement pop-up menus in apps
- implement tooltips to describe object and controls
- implement the settings menu that has control over your app
- implement custom controls in your app
- implement resource dictionaries and styles in your app
- use Generic.xaml to style custom controls
- implement various app controls inside your app

- **Lesson 4 – App Navigation**

This lesson covers the following topics:

- create basic links to navigate between pages and pass information while navigating
- implement the hub navigation pattern and describe when to use it
- add the master/details navigation pattern and describe when to use it
- implement the pivot navigation pattern and describe when it's used
- work with the nav pane navigation pattern and describe when it's used
- use a SplitView to enhance your nav pane UI
- use the back button to go back in the app navigation history
- implement keyboard navigation features
- describe the role and use of UI Automation in navigation
- define the role and use of the narrator in navigation
- implement different app navigation patterns

- **Lesson 5 – Application Life Cycle**

This lesson covers the following topics:

- describe the UWP app life cycle
- implement app activation in your app
- implement prelaunch for your app
- launch your app during Windows startup
- activate your app from a URI
- activate your app from a file
- implement suspension handling in your app
- implement app resume from suspension or termination
- implement extended execution in your app
- handle various suspension errors
- implement several app life cycle features in an app

- **Lesson 6 – Data Access and Binding in Apps**

This lesson covers the following topics:

- use x:Bind to bind data in layouts
- implement data templates with x:Bind

- work with the binding extension in your app
 - implement converting classes for the binding extension
 - add SQLite to your app's project
 - manage SQLite databases from a UWP app
 - implement CRUD operations in your UWP app
 - use EFCore to interact easily with SQLite
 - describe the MVVM pattern
 - implement Models and ViewModels in a UWP app
 - implement Views in a UWP app
 - use event bindings in a UWP app
 - implement the dispatcher in a UWP app
 - use the HttpClient class to make http requests to web services and to access cloud data
 - create JSON objects in your UWP app to use with the HttpClient
 - implement data bindings in your app with {x:Bind} and {Binding}
- **Lesson 7 – Using Mouse, Touch, and Keyboard Input**

This lesson covers the following topics:

 - retrieve input device capabilities to identify input devices
 - work with various types of touch events
 - handle various types of mouse events
 - work with various types of keyboard events
 - implement access keys in a keyboard-enabled app
 - apply custom keyboard interactions for advanced functionality in apps
 - make an adaptive UI that can shrink to fit an onscreen virtual keyboard
 - change virtual keyboard types with the InputScope property
 - implement various types of input event handlers
- **Lesson 8– Using Other Device Input Methods**

This lesson covers the following topics:

 - add capabilities to apps manually or using the manifest designer
 - use microphone capabilities for audio input
 - use your app's enterprise authentication capabilities
 - use location capabilities to get a device's location
 - take input from inking pens
 - recognize ink input as a text or shape input
 - store drawn images in your app
 - use InkToolbar to introduce additional pen functionality
 - use Windows's built-in camera interface to capture pictures and video
 - use MediaCapture to build custom camera interfaces
 - detect a device's cameras, camera capabilities, and camera information

- implement a camera preview in your interface
- control a camera's capture properties manually
- create a simple camera app

- **Lesson 9 – Speech Input and Cortana**

This lesson covers the following topics:

- use the speech synthesizer to create basic TTS functionality
- describe SSML and its different components
- use the speech synthesizer with SSML to create advanced TTS functionality
- implement speech recognition functionality
- create advanced speech recognition functionality by modifying an app's grammars and supported languages
- use speech recognition for long dictated input
- create the Voice Command Definition for a Cortana-enabled app
- activate an app through a Cortana command
- use the PhraseList element to dynamically modify VCD commands based on app input and events
- execute background tasks with Cortana commands without opening an app
- use deep links to open apps through Cortana
- implement the hand-off and progress screens to use while your app processes commands
- implement completion screens to show that a command was successfully completed
- develop an error screen for when a voice command task can't be completed
- implement confirmation screens for voice commands
- configure a disambiguation screen to obtain additional information from a user when there are multiple possibilities
- implement various features that use UWP's voice functionality

- **Lesson 10 – App-to-app Communication and File Storage**

This lesson covers the following topics:

- share data with other apps using a share contract
- use a share contract to receive data from other apps
- implement clipboard sharing in your app
- implement drag-and-drop sharing in your app
- add app extensions in your app
- create a file using the StorageFolder and StorageFile objects
- write various types of data to a file
- read various types of data from a file
- read a file's properties
- use a file picker to open and save files

- implement data roaming to sync data across devices using key/value pairs
- use data roaming to sync data across devices using unstructured files
- use a share contract to send data to another app

- **Lesson 11 – Notifications with Toasts and Tiles**

This lesson covers the following topics:

- create and send toasts in an app using XML payloads
- install the UWP notifications NuGet package to automatically generate XML payloads from C# code
- create toasts with adaptive content
- create interactive toasts
- manage app activations when a toast is clicked in the foreground or background
- configure the look and feel of an app's primary tile
- use secondary tiles for extra content and to activate the app
- create adaptive tiles that contain extra information
- implement notifications with local tiles in an app
- implement badge notifications for an app
- configure an app to send notifications using various methods

- **Lesson 12 – Background Tasks and Reusable Components**

This lesson covers the following topics:

- implement background tasks in an app
- register background tasks with Windows 10
- implement background task app triggers
- work with conditional background tasks
- implement background task monitoring
- handle events created by background tasks
- call background tasks manually from Visual Studio in order to debug them
- configure communication between background tasks and an app
- create an app service project in Visual Studio and specify the AppService extension
- implement an app service
- deploy an app service
- implement an app client that calls an app service
- implement an app service to run the same process as the foreground app as a background task
- create a reusable Windows Runtime component
- create a Class Library
- integrate Windows Runtime components and Class Libraries in a UWP app
- implement background tasks in an app

- **Lesson 13 – Authentication and Identity Management**

This lesson covers the following topics:

- use the WebAuthenticationBroker to authenticate with a third-party application
- use cookies to implement SSO with the WebAuthenticationBroker
- set up an Azure Active Directory to be used by an app
- implement Azure Active Directory authentication
- use the Credential Locker to remember login information
- describe Windows Hello capabilities in Windows 10
- create an app that can detect if Windows Hello is enabled
- implement Windows Hello in an app's login screen
- configure an app so previous users are remembered for easy login
- implement user registration functionality with Windows Hello
- create a server that provides server-side logic for a Windows Hello app
- use server-side services in a Windows Hello app
- use the WebAuthenticationBroker to authenticate with a server-side service