

# Xamarin Mobile Professional Certification Study Guide

---

**V2.0.0 2018-03-06**

## Introduction to Xamarin.Android [AND101]

Create a Xamarin.Android project

- Choose a Xamarin.Android template to create a new app
- Create a new project in your IDE

Decompose an app into Activities

- Define the concept of an Activity
- Decompose an app into Activities

Build an Activity's UI

- Add Views to a Layout in XML
- Use the Designer tool

Write an Activity's behavior

- Designate a Main Activity
- Load an Activity's UI
- Access Views from code

Update your Android SDK

- Understand the Xamarin.Android development process
- Update your Android Tools
- Update your Android Platform SDK

## Activities and Intents [AND102]

Start an Activity in your .apk

- Create an explicit Intent
- Start an Activity

Finish an Activity

- Understand Stack Navigation
- See the behavior of the Back-button
- Programmatically finish an Activity

Pass arguments to an Activity

- Load a Bundle of arguments into an Intent
- Retrieve the arguments in the target Activity

### Get Activity results

- Pass a request code
- Return a result code and Bundle
- Retrieve results

### Start a system Activity

- Create an implicit Intent
- Load Intent Action, Data, and Extras
- Verify that Android found an Activity that matches your implicit Intent

## Introduction to Xamarin.iOS [IOS101]

### Introduce the development tools

- Explore the IDE choices
- Create an app using the project templates

### (De)constructing the application

- Explore a new project
- Model-View-Controller
- Delegates and Protocols

### Add views and behavior

- Create screens
- Manage parent-child relationships
- Position views
- Add behavior

## Introduction to the Xamarin Designer for iOS [IOS102]

### Create a single screen application

- Describe the iOS Designer
- Identify controls and properties
- Demonstrate the designer workflow
- Lay out subviews

### Describe and use Auto Layout

- Describe the Auto Layout system
- Identify constraints
- Add constraints using the Designer

## Interact with designer-defined views programmatically

- Describe the Auto Layout system
- Identify constraints
- Add constraints using the Designer

## Navigate between view controllers

- Present a view controller
- Dismiss a view controller programmatically
- Use segues to perform navigation

## Introduction to Cross-platform Mobile Development [XAM110]

### Work with shared components

- Add NuGet packages to your application

### Share code using Shared Projects

- Share code across multiple projects with a shared project
- Execute platform-specific code from a Shared Project

### Share code using Portable Class Libraries

- Portable Class Libraries
- Profiles
- Handling Platform Abstractions

### Share code using .NET Standard libraries

- Create a .NET Standard library
- Select a .NET Standard target version for your library
- Use a .NET Standard library with a Xamarin app

## Introduction to Xamarin.Forms [XAM120]

### Create a single screen cross-platform application

- Compare traditional development to Xamarin.Forms
- Understand Xamarin.Forms project structure
- Use application components
- Create a Xamarin.Forms app

### Arrange the UI using Layouts

- Choose a layout container to structure your UI
- Add views to a layout container

## Use platform-specific features in shared code

- Change the UI per-platform
- Use platform-specific features
- Use the DependencyService

## XAML in Xamarin.Forms [XAM130]

### Examine XAML syntax

- Choose between XAML and C# to define your UI
- Define a UI in Xamarin.Forms using XAML

### Add Behavior to XAML-based pages

- Access XAML defined elements in the associated code-behind
- Handle events on XAML defined views

### Explore XAML capabilities

- Using device-specific values to define your app's UI
- Use Markup Extensions in XAML
- Using ContentView to share XAML across multiple Pages
- Compile XAML to improve performance
- XAML
- Advanced
- XAML

## Layout in Xamarin.Forms [XAM135]

### Specify the size of a view

- Specify preferred size of an Element
- Set layout options

### Arrange views with StackLayout

- Add views to a StackLayout in code and XAML
- Specify layout orientation
- Use Expands to request extra space

### Apply Attached Properties

- Apply an Attached Property in code
- Apply an Attached Property in XAML

### Arrange views with Grid

- Specify grid row and column sizes

- Add children to grid cells

Scroll a layout with ScrollView

- Use ScrollView to add scrolling
- Set the scroll direction

## Resources and Styles [XAM140]

Avoid duplicate XAML with Resources

- Use page-level Resources
- Dynamically update Resources

Create consistent UI with Styles

- Create and apply a Style
- Use Style inheritance to avoid repeated Setters

Make your Resources and Styles available across your entire app

- Create App.xaml
- Use application-wide resources

Apply the user's Accessibility choices with built-in Styles

- Apply a built-in Style
- Customize a built-in Style

## Consuming REST-based Web Services [XAM150]

Obtain the device's network capabilities

- Determine if the device has a connection
- Obtain the device's connection type
- Determine when network availability changes

Understand REST verbs

- Identify REST services
- Utilize URLs in REST
- Describe guidelines for using REST

Consume REST services with Xamarin

- Connect to a REST service
- Serialize data
- Send and receive data from a REST service

## Integrate with platform-specific network features

- Customize the HttpClient handler
- Leverage platform network stacks
- Use App Transport Security on iOS

## SQLite and Mobile Data [XAM160]

### Choose a data storage strategy

- Understand the data storage options available to your app
- Select a storage location
- Use the correct storage path for each platform

### Store data locally with SQLite

- Add SQLite.Net to your app
- Define SQLite table schema using attributes
- Create and connect to a SQLite database
- Perform CRUD operations against a SQLite database using SQLite.Net

### Use SQLite asynchronously

- Create an async capable database connection
- Perform CRUD operations asynchronously

## Preparing to publish your application [XAM220]

### Prepare you app for publication

- Update your app for publishing
- Use the linker to reduce app package size
- Choose a distribution strategy
- Select app stores for publication
- Choose a monetization strategy

© 2014 - 2018 Xamarin Inc., Microsoft. All rights reserved.