

App Development with Swift

Associate

Objective Domains

Individuals who earn the App Development with Swift Associate certification exemplify knowledge of key computing concepts and a solid foundation in programming with Swift. They'll demonstrate knowledge of the impact of computing and apps on society, economies, and cultures while exploring iOS app development.

Planning, Design and Theory

- 1.1 Summarize the design cycle
 - 1.1.1 Brainstorm, plan, prototype, evaluate
- 1.2 Summarize how sensitive data can be protected and compromised
 - 1.2.1 Sharing personal and application information
 - 1.2.2 Security challenges
 - 1.2.3 Legal, ethical and socioeconomic impacts

Project Navigation

- 2.1 Differentiate between basic file types
- 2.2 Recognize the assets available in a project
- 2.3 Define how assets are used
- 2.4 Import an asset to a project and use it correctly
- 2.5 Select the appropriate actions to hide or show different areas of the user interface

Interface Builder/iOS

- 3.1 Given a scenario, select the appropriate object(s) on the storyboard or the Document Outline
- 3.2 Use the Attributes inspector to non-programmatically modify the properties of objects and/or a view
- 3.3 Connect UIKit objects on storyboard to a Swift file
 - 3.3.1 Differentiate between an IBOutlet and an IBAction
 - 3.3.2 Determine when to connect an object as an IBOutlet or an IBAction
- 3.4 Programmatically modify the properties of objects and/or a view



App Development with Swift

Associate

Swift Language Usage

- 4.1 Write, call and/or evaluate the execution of functions
 - 4.1.1 Evaluate the use of argument labels, parameters and returns
- 4.2 Calculate the results when using various operators
- 4.3 Create and evaluate structures
 - 4.3.1 Declare the properties of a structure
 - 4.3.2 Initialize the properties of a structure
 - 4.3.3 Define methods
 - 4.3.4 Create an instance of a structure
 - 4.3.5 Use an instance of a structure
- 4.4 Create and manipulate arrays
 - 4.4.1 Declare and/or initialize an array with values
 - 4.4.2 Identify and/or modify an array element using its index
 - 4.4.3 Use and/or evaluate array properties and/or methods
- 4.5 Demonstrate how to control the flow of execution
 - 4.5.1 Create, analyze and predict loop structures and their results
 - 4.5.2 Create and interpret the outcome of conditional statements

- 4.6 Create, use and/or compare custom enumerations
- 4.7 Declare and/or evaluate constants and variables of different data types
 - 4.7.1 Differentiate between constants and variables
 - 4.7.2 Apply type inference
 - 4.7.3 Use explicit typing
- 4.8 Use the appropriate naming conventions
 - 4.8.1 Use appropriate camel casing
 - 4.8.2 Apply Swift identifier rules

Debugging

- 5.1 Use the Connections inspector to evaluate whether a connection error has occurred
- 5.2 Given a connection error scenario, determine a solution
- 5.3 Differentiate between syntax and run-time errors when building and running an app
- 5.4 Interpret console error messages
- 5.5 Recognize the purpose of breakpoints

