CERTIPORT®

A PEARSON VUE BUSINESS

Apple App Development with Swift

Objective Domains Crosswalk

(Certified User & Associate Exams)

App Development with Swift Certified User

App Development with Swift Certified User Objective Domain Crosswalk

OLD				Notes	NEW			
1	**Xcode Developer Tools**			No Change	1	**Xcode Developer Tools**		
	1.1	Identify and use the features of the Xcode interface				1.1	Identify and use the features of the Xcode interface	
		1.1.1	Navigate Xcode				1.1.1	Navigate Xcode
		1.1.2	Create and modify views with Interface Builder				1.1.2	Create and modify views with Interface Builder
		1.1.3	Demonstrate how to access documentation and help				1.1.3	Demonstrate how to access documentation and help
	1.2	Demonstrate how to build and run an app				1.2	Demonstrate how to build and run an app	
		1.2.1	on the iOS simulator				1.2.1	on the iOS simulator
		1.2.2	on the iOS device				1.2.2	on the iOS device
	1.3	Use debugging techniques to resolve errors		Added clarity		1.3	Use debugging techniques including, but not limited to, breakpoints, watchpoints, and logging to resolve errors	
		1.3.1	Set breakpoints and step through code line by line	No Change			1.3.1	Set breakpoints and step through code line by line
	1.4	Position and lay out UIKit objects		Removed to align with Swift UI update				
		1.4.1	Use auto layout					
		1.4.2	Embed objects in stack view					
		1.4.3	Use alignments and constraints					
		1.4.4	Navigate UI components via Document Outline					
		1.4.5	Implement app personality					
2	**Swift Programming Language**			No Change	2	**Swift Programming Language**		
	2.1	Declare and use basic Swift types				2.1	Declare and use basic Swift types	
		2.1.1	Describe and use data types and operators				2.1.1	Describe and use data types and operators
		2.1.2	Demonstrate the use of type casting in both safe and unsafe ways				2.1.2	Demonstrate the use of type casting in both safe and unsafe ways
		2.1.3	Demonstrate when to use constants and variables				2.1.3	Demonstrate when to use constants and variables
		2.1.4	Interpret and use basic types				2.1.4	Interpret and use basic types
	2.2	Manage data using collection types				2.2	Manage data using collection types	
		2.2.1	Arrays				2.2.1	Arrays
		2.2.2	Dictionaries				2.2.2	Dictionaries
	2.3	Know how and when to apply control flow and loops				2.3	Know how and when to apply control flow and loops	
		2.3.1	Use logical operators				2.3.1	Use logical operators
		2.3.2	Use Guard				2.3.2	Use Guard
		2.3.3	Use range operators				2.3.3	Use range operators

App Development with Swift Certified User

OLD				NOTES	NEW			
	2.4	Use functions				2.4	Use functions	
		2.4.1	Organize and structure code				2.4.1	Organize and structure code
		2.4.2	Create and call a function				2.4.2	Create and call a function
		2.4.3	Demonstrate how to use a function’s return value				2.4.3	Demonstrate how to use a function’s return value
		2.4.4	Customize internal, external, and anonymous naming of parameters in functions				2.4.4	Customize internal, external, and anonymous naming of parameters in functions
		2.4.5	Implement default parameter values				2.4.5	Implement default parameter values
	2.5	Demonstrate proper use of structs, classes and enums				2.5	Demonstrate proper use of structs, classes	
		2.5.1	Define and use properties and methods	No Change			2.5.1	Define and use properties and methods
		2.5.2	Differentiate between structures and classes				2.5.2	Differentiate between structures and classes
		2.5.3	Differentiate between various initializers				2.5.3	Differentiate between various initializers
		2.5.4	Define and use property observers				2.5.4	Define and use property observers
	2.6	Demonstrate the use of Optional types				2.6	Demonstrate the use of Optional types	
		2.6.1	Demonstrate how to unwrap Optionals safely				2.6.1	Demonstrate how to unwrap Optionals safely
		2.6.2	Apply Optional binding and Optional chaining (including but not limited to if let, guard let)				2.6.2	Apply Optional binding and Optional chaining (including but not limited to if let, guard let)
	2.7	Evaluate variable scope and shadowing				2.7	Evaluate variable scope and shadowing	
3	**iOS UIKit**				3	**View Building with SwiftUI**		
	3.1	Create view controllers to implement app logic				3.1	Position and/or layout a single SwiftUI View with standard Views and modifiers	
	3.2	Describe the view controller lifecycle				3.2	Create multiple Views to implement app logic	
	3.3	Use segues to link view controllers to prepare for, pass data, and unwind segues				3.3	Use List Views to iterate through collections	
		3.3.1	Differentiate between types of segues			3.4	Extract Subviews to simplify the structure of an overlarge View	
	3.4	Create a multi-view app with navigation hierarchy				3.5	Create a multi-view app with navigation Stacks, Links, and/or Sheets	
		3.4.1	Create and use Navigation controller	Updated to align with Swift UI update		3.6	Use @State, @Binding, @Environment, and/or @Observable to share data between Views	
		3.4.2	Create and use Tab Bar controller					
	3.5	Create and manipulate UIKit objects						
		3.5.1	Use common view objects such as labels and image views					
		3.5.2	Use common controls such as buttons and text views					
		3.5.3	Demonstrate the use of IBOutlet and IBAction to connect storyboard elements to code					

App Development with Swift Associate

App Development with Swift Associate Objective Domain Crosswalk								
OLD				NOTES	NEW			
1	**Planning, Design and Theory**				1	**Planning and Design**		
	1.1	Summarize the design cycle		No Change		1.1	Summarize the design cycle	
		1.1.1	Brainstorm, plan, prototype, evaluate	No Change			1.1.1	Brainstorm, plan, prototype, evaluate
	1.2	Summarize how sensitive data can be protected and co		No Change		1.2	Summarize how sensitive data can be protected and	
		1.2.1	Sharing personal and application information	No Change			1.2.1	Sharing personal and application information
		1.2.2	Security challenges	No Change			1.2.2	Security challenges
		1.2.3	Legal, ethical and socioeconomic impacts	No Change			1.2.3	Legal, ethical and socioeconomic impacts
				Added for clarity		1.3	Assess a visual design with accessibility in mind	
2	**Project Navigation**				2	**XCode Project Navigation**		
	2.1	Differentiate between basic file types		No Change		2.1	Differentiate between basic file types	
	2.2	Recognize the assets available in a project		Updated for clarity		2.2	After an asset has been imported, recognize available assets and how they are used in a project	
	2.3	Define how assets are used		Removed to align with Swift UI Update				
	2.4	Import an asset to a project and use it correctly		Updated for clarity		2.3	Import and/or use an asset	
	2.5	Select the appropriate actions to hide or show different		Updated for clarity		2.4	Select the appropriate actions to configure different areas of the user interface	
3	**Interface Builder/iOS**			Objective reordered from 3 to 4 Swift UI Kit replaced with Swift UI	4	**View Building with Swift UI**		
	3.1	Given a scenario, select the appropriate object(s) on the storyboard or the Document Outline				4.1	Differentiate between imperative and declarative programming	
	3.2	Use the Attributes inspector to non-programmatically modify the properties of objects and/or a view				4.2	Create Content Views using Text, Image, Shape, and/or Color	
	3.3	Connect UIKit objects on storyboard to a Swift file				4.3	Implement Modifiers including, but not limited to,	
		3.3.1	Differentiate between an IBOutlet and an IBAction			4.4	Create Container Views (HStack, VStack, ZStack, Spacer) and arrange Views inside of Stack Views	
		3.3.2	Determine when to connect an object as an IBOutlet or an IBAction			4.5	Explain the View hierarchy produced by a program	
	3.4	Programmatically modify the properties of objects and/or a view				4.6	Create and/or apply Interactive Views including, but not limited to, Button, TextField, Slider, and Toggle	
						4.7	Use @State Property Wrapper to control the appearance of a View	

App Development with Swift Associate

OLD				NOTES	NEW			
4	**Swift Language Usage**			Objective renumbered from 4 to 3. No Change to objectives	3	**Swift Language Usage**		
	4.1	Write, call and/or evaluate the execution of functions				3.1	Write, call and/or evaluate the execution of functions	
		4.1.1	Evaluate the use of argument labels, parameters and returns				3.1.1	Evaluate the use of argument labels, parameters and returns
	4.2	Calculate the results when using various operators				3.2	Calculate the results when using various operators	
	4.3	Create and evaluate structures				3.3	Create and evaluate structures	
		4.3.1	Declare the properties of a structure				3.3.1	Declare the properties of a structure
		4.3.2	Initialize the properties of a structure				3.3.2	Initialize the properties of a structure
		4.3.3	Define methods				3.3.3	Define methods
		4.3.4	Create an instance of a structure				3.3.4	Create an instance of a structure
		4.3.5	Use an instance of a structure				3.3.5	Use an instance of a structure
	4.4	Create and manipulate arrays				3.4	Create and manipulate arrays	
		4.4.1	Declare and/or initialize an array with values				3.4.1	Declare and/or initialize an array with values
		4.4.2	Identify and/or modify an array element using its index				3.4.2	Identify and/or modify an array element using its index
		4.4.3	Use and/or evaluate array properties and/or methods				3.4.3	Use and/or evaluate array properties and/or methods
	4.5	Demonstrate how to control the flow of execution				3.5	Demonstrate how to control the flow of execution	
		4.5.1	Create, analyze and predict loop structures and their results				3.5.1	Create, analyze and predict loop structures and their results
		4.5.2	Create and interpret the outcome of conditional statements				3.5.2	Create and interpret the outcome of conditional statements
	4.6	Create, use and/or compare custom enumerations		Removed to align with Swift UI Update				
	4.7	Declare and/or evaluate constants and variables of different data types		Objective renumbered from 4 to 3. No Change to objectives		3.6	Declare and/or evaluate constants and variables of different data types	
		4.7.1	Differentiate between constants and variables				3.6.1	Differentiate between constants and variables
		4.7.2	Apply type inference				3.6.2	Apply type inference
		4.7.3	Use explicit typing				3.6.3	Use explicit typing
	4.8	Use the appropriate naming conventions		Updated for clarity		3.7	Use the appropriate naming syntax	
		4.8.1	Use appropriate camel casing	Objective renumbered from 4 to 3. No Change to objectives			3.7.1	Use appropriate camel casing
		4.8.2	Apply Swift identifier rules				3.7.2	Apply Swift identifier rules

App Development with Swift Associate

OLD			NOTES	NEW			
5	**Debugging**			5	**Debugging**		
	5.1	Use the Connections inspector to evaluate whether a connection error has occurred	Removed to align with Swift UI Update				
	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	No Change		5.1	Differentiate between syntax and run-time errors when building and running an app	
	5.4	Interpret console error messages	Updated for clarity		5.2	Interpret error messages	
	5.5	Recognize the purpose of breakpoints	Removed to align with Swift UI Update				