

SYLLABUS OFFICIEL

Examen Swift Certified User

Développement d'applications iOS
avec Swift et SwiftUI (niveau utilisateur)

Certification : App Development with Swift Certified User

Niveau : Foundation / Entry-level | Public : Étudiants / Débutants en développement iOS

1. Présentation de la certification

L'examen **App Development with Swift Certified User (ADWS Certified User)** valide les compétences fondamentales en **développement d'applications iOS avec Swift et SwiftUI**. Cette certification atteste de votre maîtrise des concepts et pratiques essentiels qu'utilisent quotidiennement les développeurs Swift, ainsi que d'une bonne fluidité dans Xcode (éditeur de code source et éditeur d'interface).

La réussite de cet examen unique conduit à l'obtention de la certification **App Development with Swift Certified User**, reconnue internationalement et délivrée par **Apple** via Certiport (Pearson VUE). Elle constitue une première étape idéale pour les étudiants et débutants souhaitant démarrer une carrière de développeur iOS. L'examen a été mis à jour pour remplacer UIKit par **SwiftUI** dans les domaines d'objectifs.

Informations clés

Code de l'examen	ADWS Certified User (App Development with Swift Certified User)
Intitulé officiel	App Development with Swift: Certified User
Certification obtenue	App Development with Swift Certified User
Éditeur officiel	Apple Inc.
Centre de test	Certiport (Pearson VUE)
Version ciblée	Swift 5 / SwiftUI / Xcode (version actuelle 2025-2026)
Durée de l'examen	50 minutes
Nombre de questions	Environ 40 questions (QCM et tasks live-in-app)
Score requis	700 sur 1000
Prérequis recommandé	Environ 150 heures d'instruction Swift / SwiftUI / Xcode (recommandation officielle Apple-Certiport)
Langue de l'examen	Anglais (langue principale, autres langues selon disponibilité Certiport)
Âge minimum recommandé	14 ans et plus

Validité de la certification	3 ans à partir de la date de réussite (liée à la version de Xcode et Swift)
Politique de reprise	Délai d'attente de 24 heures avant la 1re reprise (voucher retake à utiliser sous 60 jours)
Modalité	En centre agréé Certiport (CATC) — OpenCertif est un centre Certiport autorisé
Badge numérique	Badge officiel Credly délivré automatiquement après réussite
Cours aligné	Develop in Swift Tutorials (gratuit sur Apple Books / Swift Playgrounds)

2. Profil du candidat

En tant que candidat à l'examen Swift Certified User, vous développez et validez des compétences fondamentales en programmation iOS. Vous êtes capable de :

- Naviguer dans Xcode et utiliser ses features (éditeurs source et Interface Builder).
- Créer et modifier des vues avec Interface Builder.
- Accéder à la documentation et à l'aide intégrées.
- Construire et exécuter une app sur l'iOS Simulator et sur un appareil iOS.
- Utiliser des techniques de debugging : breakpoints, watchpoints, logging.
- Set des breakpoints et naviguer dans le code ligne par ligne.
- Déclarer et utiliser les types de base Swift et leurs opérateurs.
- Gérer des données avec les collection types (Arrays, Dictionaries).
- Appliquer le control flow et les loops (Guard, range operators, opérateurs logiques).
- Écrire et utiliser des fonctions (paramètres nommés, valeurs par défaut, return).
- Manipuler structs et classes, properties, methods, initializers, property observers.
- Utiliser les Optional types (unwrapping, optional binding, optional chaining).
- Comprendre la portée des variables (variable scope, shadowing).
- Positionner et configurer une SwiftUI View (modifiers, layout).
- Créer des Lists SwiftUI pour itérer sur des collections.
- Créer une app multi-vues avec NavigationStack, Links et Sheets.
- Partager des données entre Views avec @State, @Binding, @Environment, Observable.

L'examen évalue spécifiquement trois familles de compétences essentielles à tout développeur iOS débutant :

- Xcode Developer Tools — interface, build/run, debugging.
- Swift Programming Language — types, collections, control flow, functions, structs/classes, optionals.
- View Building with SwiftUI — Views, modifiers, Lists, navigation, state.

3. Prérequis et public cible OpenCertif

Aucun prérequis académique formel n'est exigé pour passer l'examen Swift Certified User. Apple et Certiport recommandent toutefois :

- **Environ 150 heures d'instruction Swift / SwiftUI / Xcode** (recommandation officielle).
- Connaissance du cours gratuit **Develop in Swift Tutorials** (Apple Books / Swift Playgrounds).
- Accès à un Mac avec Xcode installé pour la pratique.
- Notions de programmation générale (variables, fonctions, types).
- Anglais niveau scolaire suffisant pour comprendre les questions de l'examen.
- Aucun prérequis Apple Developer Program n'est nécessaire.

Public cible OpenCertif

- Étudiants en informatique, multimédia, design d'applications mobiles.
- Lycéens et collégiens (à partir de 14 ans) passionnés par l'écosystème Apple.
- Auto-didactes ayant suivi les Develop in Swift Tutorials.
- Profils en reconversion vers le développement iOS.
- Candidats aux métiers de Junior iOS Developer, Mobile App Developer.
- Futurs candidats à la certification **Swift Associate** (niveau supérieur).
- Étudiants en parcours universitaires souhaitant valoriser un portfolio iOS.

4. Domaines de compétences mesurées

L'examen est structuré autour de 3 grands domaines de compétences. Le tableau ci-dessous indique le poids relatif de chaque domaine dans l'évaluation finale (version 2026 — aligné sur Swift 5 / SwiftUI / Xcode). Les pondérations sont des estimations issues du guide officiel Unity / Certiport.

Domaine	Intitulé	Pondération
1	Xcode Developer Tools	15 — 20 %
2	Swift Programming Language	45 — 55 %
3	View Building with SwiftUI	30 — 35 %

Remarque : l'examen UCU Programmer dure environ 50 minutes pour 40 questions, soit environ 1 minute 15 par question. La gestion du temps est essentielle. Le score requis pour valider est de **500 sur 700** (sur une échelle officielle Unity de 200 à 700 points).

5. Détail des compétences mesurées

Cette section détaille de manière exhaustive l'ensemble des compétences couvertes par l'examen Swift Certified User, en s'appuyant sur les Objective Domains publiés par Certiport et Unity Technologies (version 2026 — aligné sur Swift 5 / SwiftUI / Xcode).

1 Xcode Developer Tools

15 — 20 %

1.1 Interface Xcode et Interface Builder

- ▶ Identifier et utiliser les features de l'interface Xcode.
- ▶ Naviguer Xcode (Navigator, Editor, Inspector, Library, Debug area).
- ▶ Créer et modifier des vues avec Interface Builder.
- ▶ Accéder à la documentation et à l'aide intégrées (Quick Help, doc Apple).

1.2 Build et exécution d'app

- ▶ Construire et exécuter une app sur l'**iOS Simulator**.
- ▶ Construire et exécuter une app sur un **appareil iOS physique**.
- ▶ Comprendre les schémas (schemes) et configurations de build.

1.3 Debugging

- ▶ Utiliser des techniques de debugging pour résoudre les erreurs.
- ▶ Utiliser les **breakpoints**, **watchpoints** et **logging** (print, NSLog).
- ▶ Set des breakpoints et naviguer dans le code ligne par ligne (step in/over/out).
- ▶ Lire les messages d'erreur de la console et localiser leur origine.

2 Swift Programming Language

45 — 55
%

2.1 Types Swift de base

- ▶ Déclarer et utiliser les data types : Int, Double, String, Bool, Character.
- ▶ Utiliser les opérateurs : arithmétiques, comparaison, assignation.
- ▶ Démontrer le type casting (saür et non sûr : as, as?, as!).
- ▶ Démontrer quand utiliser **let** (constants) et **var** (variables).
- ▶ Interpréter et utiliser les types de base correctement.

2.2 Collection types

- ▶ Manipuler les **Arrays** (création, indexation, méthodes, propriétés).
- ▶ Manipuler les **Dictionaries** (keys/values, accès, modification).
- ▶ Comprendre les différences entre Array, Dictionary, Set.

2.3 Control flow et loops

- ▶ Utiliser les **opérateurs logiques** (&&, ||, !).
- ▶ Utiliser **guard** pour early exit.
- ▶ Utiliser les **range operators** (... ..<).
- ▶ Créer des loops : for-in, while, repeat-while.
- ▶ Créer des conditional statements : if/else, switch.

2.4 Fonctions

- ▶ Organiser et structurer le code avec des fonctions.
- ▶ Créer et appeler une fonction.
- ▶ Utiliser la valeur de retour d'une fonction (return).
- ▶ Personnaliser les labels internes, externes et anonymes des paramètres.
- ▶ Implémenter des **default parameter values**.

2.5 Structs et Classes

- ▶ Définir et utiliser properties (stored, computed) et methods.
- ▶ Différencier **structures** et **classes** (value vs reference types).
- ▶ Différencier les types d'initializers (designated, convenience, failable).
- ▶ Définir et utiliser des **property observers** (willSet, didSet).

2.6 Optional types

- ▶ Comprendre le concept d'Optional et le nil.
- ▶ Unwrapper des Optionals en toute sécurité.
- ▶ Appliquer l'**Optional binding** (if let, guard let).
- ▶ Appliquer l'**Optional chaining** (?).
- ▶ Utiliser le nil-coalescing operator (??).

2.7 Variable scope et shadowing

- ▶ Évaluer la portée d'une variable (lexical scoping).
- ▶ Comprendre le shadowing dans les closures et méthodes.

3 View Building with SwiftUI

30 — 35
%

3.1 Positionnement et layout d'une SwiftUI View

- ▶ Positionner / layout une **SwiftUI View unique** avec les Standard Views.
- ▶ Appliquer les **modifieurs** : .padding, .background, .frame, .foregroundColor, .font, .resizable.
- ▶ Utiliser Text, Image, Shape, Color.
- ▶ Organiser avec HStack, VStack, ZStack, Spacer.

3.2 Multiples Views pour la logique applicative

- ▶ Créer multiple Views pour implémenter la logique de l'app.
- ▶ Composer des Views réutilisables.

3.3 List Views

- ▶ Utiliser **List** pour itérer sur une collection.
- ▶ Configurer ForEach avec un id stable.
- ▶ Ajouter du contenu dynamique à une List.

3.4 Extraction de Subviews

- ▶ Extraire des Subviews pour simplifier la structure d'une View trop chargée.
- ▶ Refactor de View en sous-composants.

3.5 App multi-vues et navigation

- ▶ Créer une app multi-vues avec **NavigationStack**.
- ▶ Utiliser **NavigationLink** pour la navigation entre vues.
- ▶ Utiliser **Sheets** (.sheet) pour la présentation modale.

3.6 Partage de données entre Views

- ▶ Utiliser **@State** pour l'état local d'une View.
- ▶ Utiliser **@Binding** pour passer un état éditable.
- ▶ Utiliser **@Environment** pour accéder aux valeurs d'environnement.
- ▶ Utiliser **@Observable** (macro Swift) pour les objets observables.

6. Modalités pédagogiques OpenCertif

OpenCertif accompagne les candidats au Swift Certified User à travers un parcours blended-learning complet, combinant ressources e-learning interactives, projets pratiques en Swift, SwiftUI, Xcode, iOS Simulator, debugging, Views, State et UINavigationController et accompagnement tutoré.

Format de la formation

Durée recommandée	150 heures de pratique Swift et SwiftUI (recommandation officielle Apple — OpenCertif structure ce parcours sur 60 à 80 heures de formation tutorée complétées par 70 à 90 heures de projet personnel)
Modalité	100 % distanciel asynchrone, ou blended (distanciel + classes virtuelles)
Support pédagogique	Unity Certified User Courseware officiel (GMetrix) + ressources OpenCertif (modules Rise 360, scénarios immersifs)
Plateforme LMS	lmsopencertif.fr (Moodle) — accès 24/7 pendant 12 mois
Encadrement	Tutorat asynchrone par expert Unity certifié + classes virtuelles bimensuelles
Pratique requise	Au moins 150 heures de pratique Unity (recommandation officielle Unity Technologies)
Évaluations	Quiz formatifs par module, 3 projets pratiques Unity, examens blancs CertPREP
Certification finale	Passage de l'examen Swift Certified User en centre OpenCertif (CATC Certiport)

Parcours d'apprentissage proposé

- **Module 1** : Découverte de Xcode — interface, navigation, projet iOS.
- **Module 2** : Build / run sur Simulator et appareil iOS.
- **Module 3** : Debugging — breakpoints, watchpoints, logging.
- **Module 4** : Swift fondamentaux — data types, opérateurs.
- **Module 5** : Constants vs variables, type casting et type inference.
- **Module 6** : Collections — Arrays, Dictionaries.
- **Module 7** : Control flow — if/else, switch, guard, loops.
- **Module 8** : Fonctions — paramètres, labels, default values, return.

- **Module 9** : Structs et Classes — properties, methods, initializers.
- **Module 10** : Property observers (willSet, didSet).
- **Module 11** : Optionals — unwrap, binding, chaining, nil-coalescing.
- **Module 12** : Variable scope et shadowing.
- **Module 13** : Introduction à SwiftUI — Views et modifiers.
- **Module 14** : Layout SwiftUI — HStack, VStack, ZStack, Spacer.
- **Module 15** : Multiple Views et logique applicative.
- **Module 16** : Listes SwiftUI — List, ForEach.
- **Module 17** : Extraction de Subviews et refactoring.
- **Module 18** : Navigation — NavigationStack, NavigationLink, Sheets.
- **Module 19** : State management — @State, @Binding, @Environment.
- **Module 20** : Mini-projet final — app iOS multi-écrans.
- **Module 21** : Examen blanc CertPREP et préparation finale.

7. Ressources d'étude officielles

En complément du parcours OpenCertif, les ressources officielles Unity Technologies et Certiport suivantes sont fortement recommandées :

- **Develop in Swift Tutorials** (Apple Books / Swift Playgrounds) — cours officiel aligné sur l'examen.
- Swift documentation officielle Apple (swift.org / developer.apple.com/swift).
- SwiftUI documentation officielle (developer.apple.com/documentation/swiftui).
- Xcode documentation (developer.apple.com/documentation/xcode).
- App Development with Swift Certified User Objective Domains (PDF officiel Certiport).
- CertPREP Practice Tests — examens blancs Certiport alignés sur ADWS Certified User.
- WWDC sessions Apple sur SwiftUI et Swift.
- Apple Developer Academy et Apple Education Community.
- Page Certiport officielle : certiport.pearsonvue.com/Certifications/Apple/App-Dev-With-Swift.
- Page officielle Apple : certifiedprofessional.adobe.com (FAQ) et certiport.com/apple.
- Badge officiel délivré via Credly (credly.com).
- Pages OpenCertif dédiées : opencertif.fr/apple-swift.

8. Modalités de passage de l'examen

Inscription	Via OpenCertif ou directement auprès d'un centre Certiport
Centre d'examen	OpenCertif — Centre Certiport Authorized Testing Center (CATC) / Pearson VUE
Mode de passage	En centre uniquement (Unity n'autorise pas l'examen OnVUE à distance pour les certifications UCU — présence sur site requise)
Pièce d'identité	1 pièce d'identité avec photo obligatoire le jour de l'examen (pour les mineurs : autorisation parentale et CNI / passeport)
Aménagements	Demande possible auprès de Certiport (temps additionnel, assistance technique)
Résultat	Score communiqué immédiatement à la fin de l'examen (échelle 200-700, seuil de réussite 500)
Validité de la certification	3 ans à partir de la date de réussite — attribuée une seule fois (stackable, pas de renouvellement payant requis)

Politique de reprise	Délai d'attente de 24 heures avant la 1re reprise. Voucher retake à utiliser sous 60 jours après l'échec.
Badge numérique	Badge officiel délivré via Credly et intégrable à LinkedIn, CV, portfolio, sites de recrutement

9. Contact et inscription

Pour toute information complémentaire, demande de devis ou inscription à la formation préparatoire au Swift Certified User, l'équipe OpenCertif reste à votre disposition. OpenCertif est un Centre Certiport Authorized Testing Center (CATC) habilité à délivrer les certifications Unity Certified User.



10. Mentions légales et version

Ce syllabus est établi par OpenCertif sur la base des Objective Domains officiels publiés par Certiport pour la certification Swift Certified User, dans sa version applicable (version 2026 — aligné sur Swift 5 / SwiftUI / Xcode). Les compétences mesurées, les pondérations et les objectifs présentés reflètent fidèlement la structure de l'examen telle que publiée par Unity Technologies via Certiport.

Apple, le logo Apple, Xcode, Swift, SwiftUI, iOS, iPhone, iPad et le App Store sont des marques déposées d'Apple Inc., enregistrées aux États-Unis et dans d'autres pays. Develop in Swift et Swift Playgrounds sont des marques déposées d'Apple Inc. Certiport et CertPREP sont des marques déposées de Pearson Education Inc. Pearson VUE est une marque déposée de Pearson Education Inc. GMetrix est une marque déposée de GMetrix LLC. Credly est une marque déposée de Pearson Education Inc.

OpenCertif n'est pas affilié à Unity Technologies. Ce document est fourni à titre informatif. Pour la version officielle et à jour des Objective Domains, consulter certiport.pearsonvue.com/Certifications/Unity et unity.com/products/unity-certifications.

Version du syllabus : 2026.05 — Édition mai 2026

Source officielle Certiport : certiport.pearsonvue.com/Certifications/Unity/Certified-User/Certify

Source officielle Unity : unity.com/products/unity-certifications/user-programmer

Page OpenCertif : opencertif.fr/unity-user-programmer