

SYLLABUS OFFICIEL

Examen Swift Associate

Développement avancé
avec Swift et iOS (niveau associé)

Certification : App Development with Swift Associate

Niveau : Associate / Post-secondary | Public : Étudiants post-secondaires / Futurs iOS
developers

1. Présentation de la certification

L'examen **App Development with Swift Associate (ADWS Associate)** valide les compétences en **développement iOS et programmation Swift au niveau post-secondaire**. Cette certification atteste d'une connaissance des concepts clés du computing, des fondations solides de la programmation Swift, et de la compréhension de l'impact des apps sur la société, les économies et les cultures.

La réussite de cet examen unique conduit à l'obtention de la certification **App Development with Swift Associate**, conçue pour préparer les candidats post-secondaires à une carrière en développement iOS. Elle valide la connaissance de Swift et SwiftUI, et prépare aux opportunités d'études supérieures et de carrière en app development et coding.

Informations clés

Code de l'examen	ADWS Associate (App Development with Swift Associate)
Intitulé officiel	App Development with Swift: Associate
Certification obtenue	App Development with Swift Associate
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 Associate, vous développez et validez des compétences en développement iOS de niveau post-secondaire. Vous êtes capable de :

- Résumer le design cycle : brainstorm, plan, prototype, evaluate.
- Résumer comment les données sensibles peuvent être protégées et compromises.
- Identifier les challenges de sécurité et les impacts légaux, éthiques, socio-économiques.
- Différencier les basic file types Xcode (.swift, .storyboard, .xib, .plist, .json).
- Reconnaître et importer correctement les assets dans un projet.
- Sélectionner les actions appropriées pour montrer/cacher les zones de l'UI.
- Sélectionner les objets appropriés sur le storyboard / Document Outline.
- Utiliser l'Attributes Inspector pour modifier non-programmatiquement les propriétés.
- Connecter des objets UIKit ou SwiftUI à un fichier Swift.
- Différencier un **IBOutlet** et un **IBAction**, et savoir quand utiliser chacun.
- Modifier programmatiquement les propriétés des objets et des vues.
- Écrire, appeler et évaluer l'exécution de fonctions (argument labels, parameters, returns).
- Calculer les résultats lors de l'utilisation de divers opérateurs.
- Créer et évaluer des structures (properties, initializers, methods, instances).
- Créer et manipuler des arrays (declarer, modifier, propriétés, méthodes).
- Démontrer le contrôle du flux d'exécution (loops, conditional statements).
- Créer et comparer des **custom enumerations**.
- Appliquer les conventions de nommage Swift (camelCase, identifier rules).
- Debugger les erreurs de connexion, distinguer syntax vs run-time errors.
- Interpréter les messages d'erreur de la console et utiliser les breakpoints.

L'examen évalue spécifiquement cinq familles de compétences essentielles :

- Planning, Design and Theory — design cycle, security, ethics.
- Project Navigation — file types, assets, UI areas.
- Interface Builder / iOS — storyboard, IBOutlet, IBAction.
- Swift Language Usage — functions, operators, structs, arrays, control flow, enumerations.
- Debugging — connection errors, syntax vs runtime, breakpoints.

3. Prérequis et public cible OpenCertif

Aucun prérequis académique formel n'est exigé, mais Apple et Certiport recommandent :

- **200 à 250 heures d'instruction Swift / SwiftUI / Xcode** (recommandation post-secondaire).
- Recommandé d'avoir préalablement obtenu la certification **Swift Certified User**.
- Accès à un Mac avec Xcode pour la pratique.
- Connaissance fondamentale de Swift et SwiftUI.

- Familiarité avec le cours **Develop in Swift** (Tutorials, Fundamentals, Data Collections).
- Notions de design d'interface et UX iOS.
- Anglais niveau scolaire suffisant pour comprendre les questions de l'examen.

Public cible OpenCertif

- Étudiants en BTS / DUT / Licence informatique ou design d'apps.
- Étudiants en Apple Developer Academy et programmes post-secondaires Apple.
- Auto-didactes ayant terminé plusieurs projets iOS personnels.
- Junior iOS developers cherchant à valider leurs compétences.
- Profils en reconversion vers le développement mobile.
- Candidats aux métiers de Junior iOS Developer, Mobile App Developer, Junior SwiftUI Developer.
- Profils ayant passé le Swift Certified User et souhaitant monter en niveau.

4. Domaines de compétences mesurées

L'examen est structuré autour de 5 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	Planning, Design and Theory	10 — 15 %
2	Project Navigation	10 — 15 %
3	Interface Builder / iOS	20 — 25 %
4	Swift Language Usage	40 — 45 %
5	Debugging	10 — 15 %

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 Associate, en s'appuyant sur les Objective Domains publiés par Certiport et Unity Technologies (version 2026 — aligné sur Swift 5 / SwiftUI / Xcode).

1 Planning, Design and Theory

10 — 15 %

1.1 Design cycle

- ▶ Résumer le design cycle : **brainstorm, plan, prototype, evaluate**.
- ▶ Comprendre l'itération et le feedback utilisateur.

1.2 Protection et compromission des données sensibles

- ▶ Comprendre le partage d'informations personnelles et applicatives.
- ▶ Identifier les **challenges de sécurité** mobiles.
- ▶ Comprendre les impacts **légaux, éthiques et socio-économiques**.

2 Project Navigation

10 — 15
%

2.1 Types de fichiers

- ▶ Différencier les basic file types Xcode (.swift, .storyboard, .xib, .plist, .json, .xcassets).

2.2 Assets du projet

- ▶ Reconnaître les assets disponibles dans un projet (Asset Catalog).
- ▶ Définir comment les assets sont utilisés.
- ▶ Importer un asset dans un projet et l'utiliser correctement.

2.3 Interface Xcode

- ▶ Sélectionner les actions appropriées pour montrer / cacher les zones de l'UI.
- ▶ Naviguer entre Navigator, Editor, Inspector, Debug area.

3 Interface Builder / iOS

20 — 25
%

3.1 Storyboard et Document Outline

- ▶ Dans un scénario donné, sélectionner les objets appropriés sur le storyboard ou Document Outline.
- ▶ Comprendre la hiérarchie des views et view controllers.

3.2 Attributes Inspector

- ▶ Utiliser l'**Attributes Inspector** pour modifier non-programmatiquement les propriétés.
- ▶ Configurer les propriétés visuelles : color, font, alignment, image.

3.3 IBOutlet et IBAction

- ▶ Connecter des objets UIKit/SwiftUI sur storyboard à un fichier Swift.
- ▶ Différencier un **IBOutlet** (référence) et une **IBAction** (méthode).
- ▶ Déterminer quand connecter un objet comme IBOutlet ou IBAction.

3.4 Modification programmatique des propriétés

- ▶ Modifier programmatiquement les propriétés des objets et des vues.
- ▶ Accéder et modifier les view properties depuis le code Swift.

4 Swift Language Usage

40 — 45
%

4.1 Fonctions

- ▶ Écrire, appeler et évaluer l'exécution de fonctions.
- ▶ Évaluer l'usage des **argument labels**, parameters et returns.

4.2 Opérateurs

- ▶ Calculer les résultats lors de l'utilisation de divers opérateurs.
- ▶ Opérateurs arithmétiques, comparaison, logiques, range.

4.3 Structures

- ▶ Créer et évaluer des **structures**.
- ▶ Déclarer les propriétés d'une structure.
- ▶ Initialiser les propriétés d'une structure.
- ▶ Définir des methods sur la structure.
- ▶ Créer et utiliser une instance de structure.

4.4 Arrays

- ▶ Créer et manipuler des arrays.
- ▶ Déclarer et / ou initialiser un array avec des valeurs.
- ▶ Identifier et / ou modifier un élément d'array via son index.
- ▶ Utiliser et / ou évaluer les **array properties** (count, isEmpty) et **methods** (append, remove, sorted).

4.5 Control flow

- ▶ Créer, analyser et prédire les loop structures et leurs résultats (for-in, while, repeat-while).
- ▶ Créer et interpréter l'outcome de conditional statements (if/else, switch).

4.6 Custom enumerations

- ▶ Créer, utiliser et comparer des **custom enumerations**.
- ▶ Comprendre les raw values et associated values.

4.7 Constants, variables et types

- ▶ Déclarer et évaluer constants et variables de différents data types.
- ▶ Différencier **constants (let)** et **variables (var)**.
- ▶ Appliquer le **type inference**.
- ▶ Utiliser l'**explicit typing**.

4.8 Naming conventions

- ▶ Appliquer le **camelCase** approprié.
- ▶ Appliquer les **Swift identifier rules**.

5 Debugging

10 — 15
%

5.1 Connections Inspector

- ▶ Utiliser le **Connections Inspector** pour évaluer si une connection error a eu lieu.
- ▶ Identifier les outlets non connectés ou en double.

5.2 Résolution de connection errors

- ▶ Dans un scénario de connection error, déterminer une solution.
- ▶ Re-connecter, supprimer et reconnecter les IBOutlet/IBActions.

5.3 Syntax vs runtime errors

- ▶ Différencier **syntax errors** et **runtime errors** lors du build et de l'exécution.
- ▶ Identifier les erreurs de compilation Swift.

5.4 Messages de console

- ▶ Interpréter les messages d'erreur de la console.
- ▶ Lire la stack trace pour localiser l'origine d'un crash.

5.5 Breakpoints

- ▶ Reconnaître le but des **breakpoints**.
- ▶ Set et gérer des conditional breakpoints.

6. Modalités pédagogiques OpenCertif

OpenCertif accompagne les candidats au Swift Associate à travers un parcours blended-learning complet, combinant ressources e-learning interactives, projets pratiques en Swift, SwiftUI, Xcode, Interface Builder, structs et classes, design cycle, security et debugging et accompagnement tutoré.

Format de la formation

Durée recommandée	200 à 250 heures d'instruction Swift et SwiftUI (OpenCertif structure ce parcours sur 90 à 110 heures de formation tutorée complétées par 100 à 140 heures de projet iOS 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 Associate en centre OpenCertif (CATC Certiport)

Parcours d'apprentissage proposé

- **Module 1** : Design cycle iOS — brainstorm, plan, prototype, evaluate.
- **Module 2** : Données sensibles — sécurité, légal, éthique.
- **Module 3** : File types et asset catalog Xcode.
- **Module 4** : Navigation Xcode — montrer / cacher zones UI.
- **Module 5** : Storyboard et Document Outline.
- **Module 6** : Attributes Inspector — propriétés d'objets.
- **Module 7** : IBOutlet vs IBAction — connexions storyboard / code.
- **Module 8** : Modification programmatique des Views.

- **Module 9** : Fonctions Swift — argument labels, return.
- **Module 10** : Opérateurs et control flow.
- **Module 11** : Structures Swift — properties, methods, instances.
- **Module 12** : Arrays — manipulation et méthodes.
- **Module 13** : Loops et conditional statements.
- **Module 14** : Custom enumerations (raw / associated values).
- **Module 15** : Constants vs variables, type inference vs explicit.
- **Module 16** : Naming conventions Swift.
- **Module 17** : Debugging — Connections Inspector, breakpoints.
- **Module 18** : Syntax vs runtime errors et console.
- **Module 19** : Mini-projet 1 — app iOS avec storyboard.
- **Module 20** : Mini-projet 2 — app data-driven avec structs et arrays.
- **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 Fundamentals** et **Develop in Swift Data Collections** (Apple).
- Apple Objective Domains Associate (PDF officiel Certiport).
- Swift documentation officielle Apple.
- Documentation Xcode et Interface Builder (developer.apple.com/xcode).
- Apple Developer Academy et Apple Education Community.
- CertPREP Practice Tests — examens blancs Certiport alignés sur ADWS Associate.
- WWDC sessions Apple sur SwiftUI, Swift et debugging.
- Page Certiport officielle : certiport.pearsonvue.com/Certifications/Apple/App-Dev-With-Swift.
- Page Apple Developer : developer.apple.com/swift.
- 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 Associate, 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 Associate, 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, UIKit, iOS, iPhone, iPad, Interface Builder et le App Store sont des marques déposées d'Apple Inc., enregistrées aux États-Unis et dans d'autres pays. Develop in Swift est une marque déposée d'Apple Inc. Certiport et CertPREP sont des marques déposées de Pearson Education Inc. Pearson VUE est une marque déposée de Pearson Education Inc. 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