

SYLLABUS OFFICIEL

Examen ITS Software Development

Développement logiciel
(IT Specialist niveau fondations)

Certification : IT Specialist: Software Development

Niveau : Foundation / Entry-level | Public : Développeurs juniors / Étudiants informatique

1. Présentation de la certification

L'examen **IT Specialist: Software Development (ITS Software Development)**, délivré par **Pearson** via Certiport, valide les compétences fondamentales en les concepts fondamentaux du développement logiciel et de l'ingénierie logicielle. Cette certification atteste de votre capacité à maîtriser les concepts et pratiques essentiels du domaine, selon les bonnes pratiques de l'industrie.

La réussite de cet examen unique conduit à l'obtention de la certification **IT Specialist en Software Development**, reconnue internationalement par les employeurs et les institutions éducatives. Les ITS sont des certifications **entry-level** posées comme première brique du parcours IT, souvent validées avant les certifications constructeurs (Microsoft, Cisco, AWS, Google).

Informations clés

Code de l'examen	ITS Software Development (ITS-SoftDev)
Intitulé officiel	IT Specialist: Software Development
Certification obtenue	IT Specialist en Software Development
Technologie ciblée	Software Development
Éditeur officiel	Pearson (Pearson VUE / Certiport)
Centre de test	Certiport (Pearson VUE) — OpenCertif est centre Certiport autorisé
Niveau	Foundation / Entry-level
Programme	IT Specialist (ITS) — certifications industry-recognized
Format de l'examen	QCM scenarios + drag-and-drop + matching items + hot spot questions
Durée de l'examen	50 minutes
Nombre de questions	Environ 40 questions
Score requis	700 sur 1000 (≈ 70 %)
Prérequis recommandé	Environ 150 heures de pratique (recommandation Pearson / Certiport)

Langue de l'examen	Anglais (autres langues selon disponibilité régionale)
Âge minimum recommandé	13 ans et plus
Validité de la certification	Permanente sur la version passée (liée à la version technologique ciblée)
Politique de reprise	Délai d'attente de 24 heures avant la 1re reprise (voucher retake à utiliser sous 60 jours)
Modalité	En centre Certiport agréé (CATC) avec Compass — OpenCertif est centre Certiport autorisé
Badge numérique	Badge officiel délivré via Credly après réussite
Position dans le catalogue	Brique fondamentale du parcours IT — souvent passée avant les certifications constructeurs (Microsoft, Cisco, AWS, etc.)

2. Profil du candidat

En tant que candidat à l'examen ITS Software Development, vous développez et validez des compétences fondamentales en software development. Vous êtes capable de :

- Comprendre le **SDLC** (Software Development Life Cycle).
- Connaître les méthodologies : **Waterfall, Agile, Scrum, Kanban**, DevOps.
- Comprendre les **structures de données** : array, list, stack, queue, tree, hash table.
- Comprendre la **complexité algorithmique** (Big O).
- Connaître les **algorithmes** : tri (bubble, merge, quick), recherche (linéaire, binaire).
- Maîtriser la **POO** : encapsulation, héritage, polymorphisme, abstraction.
- Distinguer **compilation** et **interprétation**.
- Utiliser **Git** : commit, branch, merge, push, pull.
- Comprendre les **design patterns** : Singleton, Factory, Observer.
- Effectuer du **testing** : unit, integration, system, TDD.
- Connaître les principes **SOLID** et clean code.
- Comprendre les **API** : REST, SOAP, GraphQL.
- Connaître les **databases** : relationnel vs NoSQL.
- Gérer les **exceptions** et le debugging.

L'examen évalue spécifiquement les familles de compétences essentielles à tout débutant en software development :

- SDLC et méthodologies
- Programmation : concepts et paradigmes
- Structures de données et algorithmes
- POO et design patterns
- Version control (Git)
- Testing et debugging
- Databases et APIs

3. Prérequis et public cible OpenCertif

Aucun prérequis académique formel n'est exigé. Pearson et Certiport recommandent toutefois :

- **150 heures de pratique programmation et concepts.**
- Connaissance d'au moins un langage de programmation (Python, Java, JavaScript, C#, C++).
- Accès à un IDE : **VS Code**, IntelliJ, Eclipse, PyCharm.
- Compte **GitHub** ou **GitLab**.
- Notions algorithmiques de base.
- Anglais niveau scolaire suffisant.

Public cible OpenCertif

- Développeurs juniors tous langages confondus.
- Étudiants en BTS SIO option SLAM, DUT Informatique, Bachelor développement.
- Profils en reconversion vers le développement.
- Lycéens en NSI.
- Candidats aux certifs avancées : Java OCA, Microsoft AZ-204.
- Tech leads et architectes en formation.
- Profils QA / testers cherchant un fondement dév.

4. Domaines de compétences mesurées

L'examen est structuré autour de 7 grands domaines de compétences. Le tableau ci-dessous indique le poids relatif de chaque domaine dans l'évaluation finale (version 2026 — aligné sur la version actuelle ITS Pearson). Les pondérations sont des estimations issues du guide officiel Unity / Certiport.

Domaine	Intitulé	Pondération
1	SDLC et méthodologies	15 %
2	Programmation : concepts et paradigmes	20 %
3	Structures de données et algorithmes	15 %
4	POO et design patterns	20 %
5	Version control (Git)	10 %
6	Testing et debugging	10 %
7	Databases et APIs	10 %

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 ITS Software Development, en s'appuyant sur les Objective Domains publiés par Certiport et Unity Technologies (version 2026 — aligné sur la version actuelle ITS Pearson).

1 SDLC et méthodologies

15 %

1.1 SDLC phases

- ▶ **Requirements, Design, Implementation, Testing, Deployment, Maintenance.**

1.2 Méthodologies

- ▶ **Waterfall** : séquentiel.
- ▶ **Agile** : itératif (Agile Manifesto).
- ▶ **Scrum** : sprints, daily standup, retrospective.
- ▶ **Kanban** : visual workflow, WIP limits.
- ▶ **DevOps** : CI / CD, automation.

1.3 Documentation

- ▶ User stories et acceptance criteria.
- ▶ Technical documentation : architecture, API.
- ▶ README et CHANGELOG.

2 Programmation : concepts et paradigmes

20 %

2.1 Variables et control flow

- ▶ Variables, types, opérateurs.
- ▶ Conditionnels : if / else, switch.
- ▶ Loops : for, while, do-while.

2.2 Fonctions et modules

- ▶ Déclarer et appeler des fonctions.
- ▶ Paramètres, return, recursion.
- ▶ Modules et namespaces.

2.3 Paradigmes

- ▶ **Impératif, Déclaratif, OO, Fonctionnel.**
- ▶ Compilation vs interprétation.

3 Structures de données et algorithmes

15 %

3.1 Structures de données

- ▶ Arrays, lists, linked lists.
- ▶ Stack (LIFO) et queue (FIFO).
- ▶ Hash tables / dictionaries.
- ▶ Trees : binary tree, BST, heap.

3.2 Algorithmes

- ▶ **Tri** : bubble, selection, insertion, merge, quicksort.
- ▶ **Recherche** : linéaire, binaire.
- ▶ Recursion vs iteration.

3.3 Complexité

- ▶ Big O : $O(1)$, $O(\log n)$, $O(n)$, $O(n \log n)$, $O(n^2)$, $O(2^n)$.
- ▶ Time vs space complexity.

4 POO et design patterns

20 %

4.1 Concepts POO

- ▶ **Class** et **object**.
- ▶ **Encapsulation, Héritage, Polymorphisme, Abstraction.**
- ▶ Interfaces et abstract classes.

4.2 Principes SOLID

- ▶ **S**ingle Responsibility.
- ▶ **O**pen / Closed.
- ▶ **L**iskov Substitution.
- ▶ **I**nterface Segregation.
- ▶ **D**ependency Inversion.

4.3 Design patterns

- ▶ Créationnels : **Singleton, Factory**, Builder.
- ▶ Structuraux : Adapter, Decorator, Facade.
- ▶ Comportementaux : **Observer**, Strategy, Iterator.

4.4 Clean code

- ▶ Naming conventions, DRY, KISS.
- ▶ Functions short and single-purpose.

5 Version control (Git)

10 %

5.1 Git basics

- ▶ **init, clone, add, commit, push, pull, status.**
- ▶ Staging area et working directory.
- ▶ .gitignore.

5.2 Branching et merging

- ▶ **branch, checkout / switch.**
- ▶ **merge** et conflicts.
- ▶ Branching strategies : Git Flow, GitHub Flow.

5.3 Collaboration

- ▶ GitHub, GitLab, Bitbucket.
- ▶ Pull Requests / Merge Requests.
- ▶ Code review.

6 Testing et debugging

10 %

6.1 Types de tests

- ▶ **Unit tests, Integration tests, System tests.**
- ▶ **TDD** (Test-Driven Development).
- ▶ Mocks et stubs.

6.2 Frameworks

- ▶ **JUnit** (Java), **pytest** (Python), **Jest** (JS).

6.3 Debugging

- ▶ Breakpoints et step-through.
- ▶ Stack trace analysis.
- ▶ Logging vs debugging.

7 Databases et APIs

10 %

7.1 Databases

- ▶ Relationnel vs NoSQL.
- ▶ ORM : Hibernate, Entity Framework, Sequelize.

7.2 APIs

- ▶ **REST** : ressources, méthodes HTTP, JSON.
- ▶ SOAP (legacy), **GraphQL**.
- ▶ Authentication : API keys, OAuth, JWT.

6. Modalités pédagogiques OpenCertif

OpenCertif accompagne les candidats au ITS Software Development à travers un parcours blended-learning complet, combinant ressources e-learning interactives, projets pratiques en SDLC, POO, structures de données, algorithmes, Git, testing et accompagnement tutoré.

Format de la formation

Durée recommandée	150 heures de pratique Software Development recommandées par Pearson / Certiport (OpenCertif structure ce parcours sur 50 à 70 heures de formation tutorée complétées par 80 à 100 heures de projet et exercices)
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 ITS Software Development en centre OpenCertif (CATC Certiport)

Parcours d'apprentissage proposé

- **Module 1** : SDLC et cycle complet.
- **Module 2** : Waterfall vs Agile.
- **Module 3** : Scrum — rôles, sprints, ceremonies.
- **Module 4** : Kanban et DevOps.
- **Module 5** : Variables, types, control flow.
- **Module 6** : Fonctions et paradigmes.
- **Module 7** : Structures — array, list, stack, queue.
- **Module 8** : Hash tables, trees, graphs.
- **Module 9** : Algorithmes de tri et recherche.

- **Module 10** : Big O et complexité.
- **Module 11** : POO — class, object, encapsulation.
- **Module 12** : Héritage et polymorphisme.
- **Module 13** : Principes SOLID.
- **Module 14** : Design patterns essentiels.
- **Module 15** : Clean code et best practices.
- **Module 16** : Git — commit, branch, merge.
- **Module 17** : GitHub et Pull Requests.
- **Module 18** : Unit testing et TDD.
- **Module 19** : Debugging et stack trace.
- **Module 20** : REST APIs et JSON.
- **Module 21** : Mini-projet — application avec tests.
- **Module 22** : Examen blanc CertPREP.

7. Ressources d'étude officielles

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

- Documentation officielle Software Development (sources éditeur).
- Page Certiport officielle : certiport.pearsonvue.com/Certifications/IT-Specialist.
- Page Pearson IT Specialist : [pearson.com / itspecialist](https://pearson.com/itspecialist).
- **CertPREP Practice Tests (GMetrix)** — examens blancs Certiport pour ITS.
- **LearnKey courses** pour IT Specialist — self-paced video learning.
- Tutoriels gratuits et documentation Software Development.
- Communautés : **Stack Overflow, GitHub, Reddit** (selon technologie).
- Plateformes d'apprentissage : **Coursera, edX, Udemy, Pluralsight**.
- Badge officiel délivré via **Credly** (credly.com).
- Page Pearson VUE pour la réservation : home.pearsonvue.com.
- Pages OpenCertif dédiées : opencertif.fr / [its](https://its.pearsonvue.com).

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 ITS Software Development, 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 ITS Software Development, dans sa version applicable (version 2026 — aligné sur la version actuelle ITS Pearson). Les compétences mesurées, les pondérations et les objectifs présentés reflètent fidèlement la structure de l'examen telle que publiée par Unity Technologies via Certiport.

Pearson, le logo Pearson, Pearson VUE, Certiport, CertPREP, GMetrix, Compass et IT Specialist (ITS) sont des marques déposées de Pearson Education Inc. Git est une marque déposée de Software Freedom Conservancy. GitHub est une marque déposée de Microsoft Corporation. LearnKey est une marque déposée de LearnKey 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