

Fiche du cours

50 h

Titre :

SE101 - Génie logiciel 1 (Git, TDD, CI/CD, Clean Code)

Description :

Les fondamentaux professionnels pour livrer du code fiable en équipe : Git & GitHub (branches, rebase, PR), revues de code, TDD/tests unitaires, CI (GitHub Actions) et Clean Code. Tronc commun multi-langages : pratiques comparables en Python, C++ et Java. Prérequis : savoir coder des programmes simples.

Objectifs :

- Maîtriser Git : branching model, rebase, cherry-pick, PR propres.*
- Conduire des revues de code efficaces (checklist, smells fréquents).*
- Pratiquer TDD, écrire des tests unitaires (pytest / JUnit / Catch2), mocks/fakes de base.*
- Mettre en place une CI : lint + tests, cache de dépendances, matrice (versions/langages)*.
- Appliquer Clean Code : naming, petites fonctions, gestion d'erreurs, doc minimale utile

Chapitres :

1. Git (rappels) : clone, commit, branches, rebase vs merge*
2. Workflows : GitHub Flow / GitFlow, conventions de PR, règles d'équipe*
3. Code review : checklist qualité, commentaires, etiquette*
4. TDD : cycle red-green-refactor, katas (FizzBuzz/ROM/Bank)*
5. Tests unitaires multi-langages (pytest/JUnit/Catch2), mocks*
6. CI avec GitHub Actions : pipeline lint+tests, artefacts*
7. CI avancée : matrix builds, cache, rapports de couverture*
8. Clean Code : smells courants, refactor guidé par tests*
9. Qualité statique : formatters & linters (black/flake8, clang-tidy, Checkstyle), typing (mypy/...)
10. Mini-projet SE-101 : repo public avec PR revues + pipeline vert (tests/lint/coverage)

À la fin :

Vous saurez collaborer proprement (Git/PR), sécuriser votre code par des tests automatisés, industrialiser une CI robuste, et écrire un code plus lisible et maintenable — quelle que soit la techno

(Python, C++ ou Java).