

UE Méthodes et outils pour la conception avancée



Niveau d'étude
Bac +3



ECTS
3 crédits



Crédits ECTS
Echange
3.0



Composante
UFR IM2AG
(informatique,
mathématiques
et
mathématiques
appliquées)



Période de
l'année
Printemps (janv.
à avril/mai)

- > **Langue(s) d'enseignement:** Français
- > **Méthodes d'enseignement:** En présence
- > **Forme d'enseignement :** Cours magistral
- > **Ouvert aux étudiants en échange:** Oui
- > **Crédits ECTS Echange:** 3.0
- > **Code d'export Apogée:** GBIN6U08

Présentation

Description

Dans de nombreux secteurs (transports, énergie, santé,...), concevoir le plus efficacement et sûrement possible des logiciels de qualité représente un enjeu économique ou sociétal majeur. L'objectif de cette option est de familiariser les étudiants à quelques "bonnes pratiques" liées à certains des critères de qualité de la conception logicielle : solutions pour la maintenabilité et la réutilisabilité du code, méthodes de détection de défauts et erreurs, analyse de vulnérabilité, analyse de performances, et également évaluation de la qualité des tests.

Heures d'enseignement

CM	CM	15h
TD	TD	15h

Pré-requis recommandés

Aisance à la programmation en C sous Unix.

Période : Semestre 6

Compétences visées

Savoir organiser un code de façon modulaire, lisible, documentée, évolutive et réutilisable.

Maîtriser les concepts de compilation séparée et de création/utilisation de bibliothèques, et savoir utiliser les outils associés.

Savoir concevoir et exécuter des tests et évaluer leur qualité (couverture).

Maîtriser la détection de divers types de défauts et erreurs de conception ou de programmation, par des méthodes de type exécution symbolique, analyse dynamique de la gestion mémoire.

Comprendre le concept d'analyse de vulnérabilité et utiliser des tests adaptés. Comprendre les concepts liés à l'analyse de performances (efficacité du code lui-même, impact de l'architecture matérielle l'exécutant) et savoir exploiter les outils permettant cette analyse.

Logiciels utilisés = Les travaux pratiques sont réalisés en C sous Linux, et utilisent gcc, make, doxygen, CuTest, gcov et lgov, Klee, Valgrind, AFL, gprof, perf.

Infos pratiques

Contacts

Responsable pédagogique

Laurence Pierre

✉ laurence.pierre@univ-grenoble-alpes.fr

Lieu(x) ville

› Grenoble

Campus

› Grenoble - Domaine universitaire