

CS-116

Projets en informatique (pour SV)

Gewaltig Marc-Oliver, Rougemont Jacques

Cursus	Sem.	Type
Ingénierie des sciences du vivant	BA3	Obl.

Langue d'enseignement	français
Crédits	4
Session	Hiver
Semestre	Automne
Examen	Pendant le semestre
Charge	120h
Semaines	14
Heures	4 hebdo
Cours	2 hebdo
Exercices	2 hebdo
Nombre de places	

Remarque

Cours partiellement en anglais

Résumé

On présente et on met en pratique les principes de développement en équipe d'un projet logiciel complexe en C++ à l'aide d'outils utilisés dans l'industrie. Deux exemples issus des sciences de la vie sont proposés et les étudiants développent une solution et font évoluer le projet durant le semestre

Contenu

- Programmation orientée objet et "design patterns"
- Debuggage, profilage, "refactoring"
- Gestion des erreurs, tests unitaires
- Contrôle de version distribué
- Documentation de projet, style et structure de projet
- Techniques de développement agiles

Le cours est orienté sur la pratique et sur le concept de développement "SCRUM". La première moitié du semestre est consacrée à acquérir les outils nécessaires en suivant individuellement les étapes de développement d'un projet prédéfini.

Dans la deuxième partie, un second problème est présenté et les étudiants forment des groupes de 5 à 7 personnes et définissent collectivement les buts à atteindre, les tâches spécifiques à réaliser et les critères de validation, sur la base de cycles successifs de 2 semaines:

- semaine 1 - planning: établissement du plan, distribution des tâches
- semaines paires - stand-up: évaluation des progrès, modification du plan, nouvelles tâches
- semaines impaires - demo and review: présentation des résultats, planning du prochain cycle

Le cours en auditoire sera consacré à la présentation des problèmes, à l'introduction des outils nécessaires à la progression des projets, à l'enseignement de points techniques concernant la programmation et la gestion de projets en équipe.

Les heures de TP seront consacrées au planning et stand-up, environ 1 à 2 heures de travail autonome est attendu durant la semaine.

Mots-clés

C++, programmation orientée objet, algorithmique, projet de développement

Compétences requises

Cours prérequis obligatoires

Programmation I, II

Concepts importants à maîtriser

Pratique du C++

Concepts fondamentaux de la programmation orientée-objet en C++

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Evaluer le coût d'un algorithme en temps et en espace
- Analyser l'apport de techniques algorithmiques connues
- Concevoir une application répondant à des spécifications données
- Tester l'adéquation du résultat d'un programme par rapport aux fonctionnalités visées
- Réaliser de façon autonome une application de taille moyenne au moyen du langage C++
- Utiliser des outils dédiés à la réalisation de projets (gestion de version, debuggage)
- Utiliser de façon autonome des bibliothèques connues
- Choisir ou sélectionner et interpréter la documentation informatique en ligne

Compétences transversales

- Gérer ses priorités.
- Recevoir du feedback (une critique) et y répondre de manière appropriée.
- Planifier des actions et les mener à bien de façon à faire un usage optimal du temps et des ressources à disposition.
- Utiliser une méthodologie de travail appropriée, organiser un/son travail.
- Utiliser les outils informatiques courants ainsi que ceux spécifiques à leur discipline.
- Identifier les différents rôles qui caractérisent les équipes performantes et tenir différents rôles dont un rôle de coordinateur.
- Résoudre des conflits de façon productive pour la tâche et les personnes concernées.

Méthode d'enseignement

Ex cathedra, travaux pratiques sur ordinateurs, réalisation de deux projets notés pendant 13 semaines.

Travail attendu

Participation au cours. Réalisation des projets en séances d'exercices et travail individuel durant la semaine.

Méthode d'évaluation

La note finale est un combinaison de 3 évaluations: évaluation du travail individuel dans le premier projet (50%), évaluation du deuxième projet réalisé en équipe (25%), contribution individuelle au travail d'équipe (25%).

Dans le travail en équipe, des tâches de programmation sont définies collaborativement et des points de difficulté leur sont attribués. La contribution individuelle est calculée par le nombre de points accumulés par l'étudiant durant le projet (en réalisant les tâches correspondantes).

Les critères d'évaluation des projets finaux tiennent compte de:

- historique complet sur git
- intégration et exhaustivité des tests
- qualité de la documentation, clarté du code
- qualité du code, performance du programme
- compréhension des algorithmes

Encadrement

Assistants	Oui
Forum électronique	Oui

Ressources**Service de cours virtuels (VDI)**

Oui

Bibliographie

Notes téléchargeables sur le web; livre(s) de référence indiqué(s) en début de semestre

Sites web

- <http://www.cprogramming.com/tutorial/c++-tutorial.html>
- <http://www.cplusplus.com/reference/>
- http://www.cmake.org/cmake/help/cmake_tutorial.html
- <https://www.atlassian.com/git/tutorial>
- <http://www.stack.nl/~dimitri/doxygen/manual/docblocks.html>
- <https://code.google.com/p/googletest/wiki/Primer>
- <https://www.zenhub.io/>

Liens Moodle

- <http://moodle.epfl.ch/enrol/index.php?id=7941>

Préparation pour

Programmation bio-mathématique