

3MD1520 - Programmation avancée en C++

Responsables : **Herve FREZZA-BUET**

Langues d'enseignement : **FRANCAIS**

Campus où le cours est proposé : **CAMPUS DE METZ**

Nombre d'heures d'études élèves (HEE) : **50**

Nombre d'heures présentielles d'enseignement (HPE) : **20**

Année académique : **2024-2025**

Niveau avancé : **non**

Présentation, objectifs généraux du cours :

Savoir coder efficacement un algorithme dans un langage de programmation donné suppose comprendre préalablement le modèle de calcul associé et la façon dont les instructions de ce langage sont traduites en instructions machine. Trop d'étudiants abordent encore la programmation de façon superficielle et hasardeuse, faute de connaître les notions de base nécessaires à l'écriture d'un code élégant et performant.

La force singulière du langage C++ est de permettre la production de codes compilés proches du code machine optimal tout en offrant différentes approches de programmation de haut-niveau comme le typage fort, la programmation objet, la programmation fonctionnelle et la méta-programmation (génération automatique de code à la compilation). Pour cette raison, C++ est devenu le langage incontournable pour développer des algorithmes optimisés. Son seul inconvénient tient à sa richesse qui n'a cessé de croître dans ses versions les plus récentes (C++11/14/17/20) et qui rend le langage difficile à appréhender dans sa totalité sans formation adéquate.

Ce cours est destiné aux étudiants, y compris débutants, désireux de maîtriser les différents aspects de la programmation en C++ pour pouvoir écrire du code qui concilie performance et élégance. Le cours adopte une approche bottom-up en partant des mécanismes d'exécution de programmes élémentaires pour aller progressivement vers les fonctionnalités les plus avancées du langage.

Période(s) du cours (n° de séquence ou hors séquence) :

SD9

Prérequis :

Savoir programmer (boucles, fonctions, rudiments d'approche objet) dans un langage informatique, avoir des rudiments d'architecture des ordinateurs (processeur, mémoire, etc.).

Plan détaillé du cours (contenu) :

6 séances de TP (3h chacune), et un examen individuel sur machine (2h). Le programme porte sur la chaîne de compilation, la gestion mémoire (pile et tas), les types, l'héritage, la généricité (templates), la programmation système.

Déroulement, organisation du cours :

L'objectif est de transmettre aux étudiants un réel savoir-faire de la programmation, d'une part en illustrant les notions à travers des exemples d'utilisation pertinents, d'autre part en consacrant une part importante du volume horaire aux travaux de laboratoire.

Organisation de l'évaluation :

Exercices de programmation.

Moyens :

Enseignants: Hervé Frezza-Buet, Frédéric Pennerath
TP sous PC Linux / g++
40 étudiants max.
2 étudiants par machine max.

Acquis d'apprentissage visés dans le cours :

- Savoir écrire un programme en C++ utilisant différents paradigmes de programmation comme la programmation objet, la programmation fonctionnelle et la programmation générique.
- Connaître certains aspects du langage C++ qui ont une influence déterminante sur les performances des programmes lors de leur exécution.
- Connaître les fonctionnalités offertes par les spécifications les plus récentes du langage C++ (C++11, C++14, C++17, C++20).
- Savoir utiliser un environnement de compilation et de débogage C++

Description des compétences acquises à l'issue du cours :

C6.3 Conceive, design, implement and authenticate complex software.
C6.4 Solve problems through mastery of computational thinking skills.

Bibliographie :

- Site web réalisé par les professeurs
- Effective Modern C++, Scott Meyers, 2014
- Professional C++, Marc Greddoir, 2014
- A Tour of C++, Bjarne Stroustrup, 2013