

2EL5030 - Programmer efficacement en C++

Responsables : **Frederic PENNERATH**

Langues d'enseignement : **ANGLAIS**

Type de cours : **Electif 2A**

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

Nombre d'heures d'études élèves (HEE) : **60**

Nombre d'heures présentielles d'enseignement (HPE) : **30**

Année académique : **2024-2025**

Catégorie d'électif : **Sciences fondamentales**

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.

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.

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

SG8

Prérequis :

Le cours ne suppose pas de prérequis autre qu'une exposition aux bases de la programmation dans un langage informatique quelconque.

Plan détaillé du cours (contenu) :

Volume total de 15h de cours, 18h de TP et 3h d'examen

Partie magistrale (15h) :

- Gestion mémoire et cycle de vie des variables mémoire (1h30)
- Structure et smart pointers (1h30)
- Types (1h30)
- Programmation fonctionnelle : lambda fonctions, types appelables, exceptions (1h30)
- Classe et héritage (1h30)
- Mécanisme de base des templates (1h30)
- Programmation générique à base de templates (1h30)
- Programmation générique versus programmation orientée objet (1h30)
- Bibliothèque standard : contenu et concepts (1h30)
- Programmation système : threads et mécanismes de synchronisation (1h30)

Partie pratique (18h) :

- TP sur la manipulation mémoire (3h)
- TP sur la programmation objet et l'héritage (3h)
- TP sur la programmation fonctionnelle (3h)
- TP sur la programmation générique (3h + 1h30)
- TP sur la bibliothèque standard (1h30)
- TP sur la programmation système (3h)

Examen sous forme de TP (3h)

Déroulement, organisation du cours :

- Cours basé sur des exemples de code
- Partie importante (50%) consacrée aux travaux pratiques de programmation

Organisation de l'évaluation :

L'évaluation se fera sur la base d'un seul examen prévu à la fin du cours.

Cet examen d'une durée de deux heures consistera à réaliser de courts exercices de programmations. Chaque étudiant travaillera seul sur sa machine sans avoir accès à Internet mais disposera de toute la documentation nécessaire pour réussir l'épreuve (supports de cours, documentation technique).

L'examen de rattrapage aura lieu selon les mêmes modalités que l'examen initial.

Moyens :

- Deux enseignants : Hervé Frezza-Buet et Frédéric Pennerath
- Pas de TD mais uniquement des travaux pratiques sur machine
- Un poste de travail par élève
- Un ou deux groupes de TP d'au maximum 15 étudiants chacun.
- Les TP seront réalisés dans l'environnement Linux et reposeront exclusivement sur des logiciels libres (g++, CMAKE, etc).

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 et C++20).
- Savoir utiliser un environnement de compilation et de débogage C++

Description des compétences acquises à l'issue du cours :

- C2: Develop a deep competence in an engineering field and in a family of professions.
- C6: Be operational, responsible and innovative in the digital world.

Bibliographie :

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