

# Compléments de programmation et d'algorithmique

## Titulaire

Jérémie ROLAND (Coordonnateur)

## Mnémonique du cours

INFO-H304

## Crédits ECTS

5 crédits

## Langue(s) d'enseignement

Français

## Période du cours

Premier quadrimestre

## Campus

Solbosch

## Contenu du cours

Langage C:

- > Syntaxe et sémantique
- > Allocation et désallocation manuelle de mémoire; pointeurs

Notions de C++

- > Syntaxe et sémantique, aspects de programmation orientée objet
- > Templates et bibliothèque standard C++ (STL)

Analyse de base des algorithmes

- > Les notations Grand O, Omega et Theta
- > Notion d'algorithme et de complexité
- > Classes de complexité standards

Structures de données

- > Principes généraux
- > Listes chaînées
- > Queues de priorité, tas (heaps)
- > Arbres binaires de recherche
- > Tables de hachage

Algorithmes de tri

- > Tri par insertion
- > Tri par fusion (mergesort)
- > Tri par tas (heapsort)- Tri rapide (quicksort)
- > Tri linéaire: tri par comptage (counting sort) et tri Radix

Stratégies algorithmiques

- > Divide-and-conquer

- > Backtracking
- > Programmation dynamique
- > Algorithmes gloutons

## Objectifs (et/ou acquis d'apprentissages spécifiques)

Programmation en C et C++:

- > Pouvoir écrire des programmes simples en C/C++
- > Comprendre et utiliser les concepts de base de la gestion dynamique de la mémoire

Algorithmique et structures de données:

- > Pouvoir concevoir des algorithmes pour des problèmes simples en utilisant les techniques de base de l'algorithmique (divide-and-conquer, backtracking, programmation dynamique, etc.)
- > Pouvoir concevoir et utiliser une structure de données élémentaire adaptée au problème (tableaux, listes chaînées, tables de hachage, arbres binaires de recherche, etc.)
- > Pouvoir analyser la complexité de ces algorithmes

## Pré-requis et co-requis

### Cours pré-requis

INFO-H2001 | Programmation orientée objet | 5 crédits

### Cours ayant celui-ci comme co-requis

INFO-F201 | Systèmes d'exploitation | 5 crédits

## Méthodes d'enseignement et activités d'apprentissages

- > Cours magistraux avec slides et démonstration sur ordinateur. Mise-en pratique directe sur ordinateur via une machine virtuelle. Enseignement en présentiel, retransmis en direct via Teams, et enregistré.
- > Travaux pratiques en salle informatique: programmation sur ordinateur, mais aussi certains exercices théoriques sur papier. Travaux pratiques organisés en présentiel et séances de questions-réponses via Teams
- > Projet de programmation sur un sujet imposé

## Contribution au profil d'enseignement

Cette unité d'enseignement contribue aux compétences suivantes :

- > Abstraire, modéliser et simuler des systèmes physiques complexes rencontrés dans les applications biomédicales (bioélectricité, biomécanique, écoulements, etc.)
- > Se représenter les mécanismes biologiques fondamentaux depuis la biochimie de la cellule jusqu'au fonctionnement des principaux systèmes de la physiologie humaine
- > Gérer, explorer et analyser les données médicales (dossier médical, imagerie, génomique, statistiques)

## Références, bibliographie et lectures recommandées

- > The C Programming Language (Second Edition), B.M. Kernighan and D.M. Ritchie, Prentice Hall (1988)
- > The C++ Programming Language (Fourth Edition), B. Soustrup (2013)
- > C++ for Java Programmers, M.A. Weiss (2003)
- > Introduction to Algorithms (Third Edition), T.H. Cormen et al. (2009)

## Support(s) de cours

Podcast et Université virtuelle

## Autres renseignements

### Lieu(x) d'enseignement

Solbosch

### Contact(s)

Jérémie Roland (Jeremie.Roland@ulb.be)

## Méthode(s) d'évaluation

Examen oral et Projet

### Méthode(s) d'évaluation (complément)

- > Evaluation continue: projet de programmation (5 points)
  - > Projet en groupe sur un sujet imposé
  - > Cahier des charges fourni en octobre, reprenant les critères d'évaluation

- > Echéance intermédiaire en novembre: code avec fonctionnalités de base
- > Echéance finale en décembre: code avec toutes les fonctionnalités + rapport
- > Evaluation finale: examen oral (15 points)
  - > 1 question d'algorithmique (10 points)
  - > 1 question de programmation (5 points)
  - > 45 minutes de préparation des réponses : sans accès au matériel du cours, écriture des réponses sur le tableau (pas d'ordinateur pour la partie programmation)
  - > 45 minutes de présentation des réponses

## Construction de la note (en ce compris, la pondération des notes partielles)

Le projet est noté sur 5 et l'examen oral sur 15.

- > Si les deux notes sont supérieures à la moitié, la note globale est la somme de ces deux notes.
- > Si le projet ou l'examen oral est en échec, les notes sont ramenées sur 20 et la note globale est le minimum des deux notes.

En conclusion, pour valider le cours il faut réussir le projet **et** l'examen oral.

Le report d'une note partielle (projet ou examen oral) d'une session à l'autre se fait uniquement pour les notes supérieures à la moitié ; ce report est alors automatique. Si le projet n'a pas été validé en première session, il peut être réalisé en seconde session, les étudiants concernés doivent alors contacter l'enseignant pour organiser la remise de ce projet de seconde session.

## Langue(s) d'évaluation principale(s)

Français

## Programmes

### Programmes proposant ce cours à l'école polytechnique de Bruxelles

BA-IRCI | Bachelier en sciences de l'ingénieur, orientation ingénieur civil | option Bruxelles/bloc 3