

Complément d'informatique

INFO0952

Pierre Geurts

Dernière mise à jour le 24 septembre 2018

E-mail : `p.geurts@ulg.ac.be`
URL : `http://www.montefiore.ulg.ac.be/~geurts/ci.html`
Bureau : R 134 (Montefiore)
Téléphone : 04.366.48.15

Contact

- Chargé de cours :
 - ▶ Pierre Geurts, p.geurts@uliege.be, I.134 Montefiore, 04/3664815
- Assistant :
 - ▶ Nicolas Vecoven, nvecoven@uliege.be, II.99 Montefiore
- Sites web du cours :
 - ▶ Cours théorique :
<http://www.montefiore.ulg.ac.be/~geurts/ci.html>
 - ▶ Répétitions et projets :
<http://www.montefiore.ulg.ac.be/~nvecoven/ci/ci.html>

Les objectifs du cours

- Consolider et étendre vos connaissances d'un langage de programmation (le C)
- Vous apprendre à écrire des programmes pour résoudre des problèmes réalistes (de taille moyenne)
- Vous initier à l'algorithmique et à l'étude des structures de données
- Vous ouvrir à d'autres paradigmes de programmation
- Et, on ne sait jamais, vous donner le goût pour la programmation (et l'informatique)

L'informatique dans le bachelier ingénieur

Bloc 1 :

- (Obl) INFO2009 - Introduction à l'informatique
- (Obl) INFO0061 - Organisation des ordinateurs

Bloc 2 :

- **(Obl) INFO0952 - Complément d'informatique**
- INFO0902 - Structures de données et algorithmes
- INFO0062 - Object-oriented programming

Bloc 3 :

- INFO0012 - Computation structures
- INFO0004 - Projet de programmation orientée-objet
- INFO0009 - Base de données
- INFO0054 - Programmation fonctionnelle
- INFO0010 - Introduction to computer networking
- INFO8006 - Introduction to artificial intelligence

Approche pédagogique

Apprentissage par la pratique :

- 5 ou 6 séances de travaux dirigés sur papier et/ou ordinateur encadrés par des assistants et élèves-moniteurs
- 4 petits devoirs de programmation à réaliser individuellement, lors des séances de travaux dirigés et à la maison.
- 2 projets de plus grande envergure à faire seul ou par groupe de deux, lors de séances de travaux dirigés et à la maison.

Cours théorique :

- Compléments de programmation en C, en particulier sur l'écriture et l'organisation de programmes
- Compléments d'algorithmique (programmation récursive, notions de complexité, tri...)
- Introduction aux structures de données (pile, file, liste, table de hachage...)
- Feedback général sur les devoirs et projets

Matière du cours

1. Rappel
2. Récursivité
3. Organisation de programmes
4. Complexité
5. Tri et recherche
6. Structures de données
7. Langages programmation

Organisation pratique

■ Cours théoriques :

- ▶ Les mardis de 15h00 à 17h00, Amphi A202 B7b
- ▶ 8-9 cours
- ▶ Transparents disponibles sur la page web du cours avant chaque cours

■ Travaux dirigés :

- ▶ Certains mardis de 13h00 à 15h00, par petits groupes.
- ▶ Instructeurs : professeur, assistant et élèves-moniteurs
- ▶ Exercices sur ordinateur ou sur feuille portant sur la matière théorique ou la réalisation des devoirs et projets. Énoncés disponibles sur la page web des projets.

■ Projets :

- ▶ Deux projets à réaliser pendant le semestre
- ▶ Le premier individuellement, le deuxième en binôme

Modalités d'évaluation

- Devoirs : 20% (5% par devoir)
- Projets : 60% (30% par projet)
- Examen écrit : 20% (à livre ouvert, portant sur la matière du cours, des répétitions et les projets)
- En deuxième session :
 - ▶ Même modalités et pondérations
 - ▶ les devoirs et projets non rendus ou ratés devront être refaits
- Le cours étant fortement basés sur les devoirs et projets, pas de report de cote d'une année à l'autre

Critères d'évaluation des devoirs et projets

Trois critères principaux :

- Exactitude du code : testée automatiquement, en partie lors de la soumission des projets sur la plateforme
- Style et utilisation du langage : vérification rapide, menant à un cotation (presque) binaire : ok ou pas.
- Rapport (projets uniquement) : précision et qualité des réponses

Un feedback global sur les devoirs et projets sera donné lors de certains cours théoriques. Un feedback plus spécifique pourra être obtenu sur demande.

Collaboration et plagiat

- La collaboration entre étudiants sur le cours théorique, la programmation C, l'utilisation d'outils et les exercices de répétition est fortement encouragée
- Discuter des concepts généraux liés aux devoirs et aux projets entre étudiants est permis.
- **Montrer son code à d'autres étudiants, regarder/copier, même partiellement, le code d'autres étudiants ou du code obtenu via d'autres sources est strictement interdit !**
- Des outils sophistiqués de détection de plagiat, robustes à des modifications de noms de variables ou des réarrangements de code, seront utilisés systématiquement.
- En cas de plagiat avéré (après convocation des étudiants incriminés), des sanctions seront appliquées.

Notes de cours

Les transparents des cours théoriques disponibles sur la page web du cours (un peu) avant chaque leçon.

Pas de livre de référence obligatoire mais les livres suivants ont été utilisés pour préparer le cours :

- C programming : a modern approach, K.N. King, W. W. Norton & Company, Second edition, 2008.
- Le langage C - Norme ANSI, Kernighan et Ritchie, Dunod, 2ème édition, 2000.
- Computer science : an interdisciplinary approach, Sedgewick et Wayne, Pearson, 2016.
- Introduction to algorithms, Cormen, Leiserson, Rivest, Stein, MIT press, Third edition, 2009.

Contenu du cours

- Partie 1: Rappel
- Partie 2: Récursivité
- Partie 3: Organisation de programmes
- Partie 4: Complexité
- Partie 5: Tri et recherche
- Partie 6: Structures de données
- Partie 7: Langages de programmation