

# CS444 – Langages et compilation – 2022-2023

## Description

### Descriptif du module

- Crédits ECTS : 5
- Volume : 9 CM, 8 TD, 10 TDM de 3h (projet)

### Objectifs

Le but du cours est de donner aux étudiants un aperçu du fonctionnement d'un compilateur pour un langage impératif, type Pascal ou C. Les aspects théoriques sont complétés par la mise en oeuvre de quelques techniques et outils classiques utilisés en compilation.

### Prérequis

- cours CS322 'informatique théorique' (langages, grammaires, automates)
- expérience d'au moins un langage de programmation (C, Java...)
- notions de langage assembleur

### Évaluation

- E1 = Écrit 1h30
- CC = contrôle continu (qcms)
- PR = note de projet
- $N1 = 0,45 * E1 + 0,45 * PR + 0,1 * CC$
- $N2 = 0,5 * E2 + 0,5 * PR$

### Intervenants

- CM : Christophe Deleuze (4 CM) puis Laure Gonnord (5 CM)
- TD : LG+CD
- projet : LG+CD

### Le projet

TBD L'organisation et les modalités du projets seront publiées sur chamilo.

### Bibliographie

- *Compilateurs – Principes, techniques et outils*, Aho, Sethi, Ullman. InterÉditions 1989 (orig. 1986, deuxième édition 2007)

Le "livre du dragon", écrit il y a plus de 30 ans, reste une très bonne référence. Disponible à la bibliothèque UGA.

- *Compilateurs*, Grune, Bal, Jacobs, Langendoen (Dunod 2002, orig. 2000, *Modern compiler design*, deuxième édition 2012)

Souvent d'accès plus difficile. Comme son titre original le suggère, prend en compte le traitement des langages “modernes” : fonctionnels, à objets, et distribués.

## Modalités

- les supports de cours sont placés sur chamilo
- chaque étudiant doit s'assurer qu'il est inscrit au cours sur chamilo
- la compilation est un sujet dense : il est particulièrement important de travailler très régulièrement
- du travail de préparation sera parfois demandé avant les séances de cours

## Planning du cours

- séance 1 : aperçu, analyse lexicale
- séance 2 : analyse syntaxique
- séance 3 : analyse syntaxique
- séance 4 : grammaires attribuées
- séance 5 : arbre abstrait, typage
- séance 6 : génération de code
- séance 7 : représentations intermédiaires
- séance 8 : allocation de registres
- séance 9 : analyse de flot de données