

Programmation en langage C - 2



En bref

- › **Langues d'enseignement:** Français
- › **Méthodes d'enseignement:** En présence
- › **Ouvert aux étudiants en échange:** Oui

Présentation

Description

Ce module présente les principaux aspects de la programmation en langage C, en donnant une vision précise de ses principales caractéristiques. La programmation impérative et structurée est présentée en détails, d'une part au travers du découpage en fonction, d'autre part par la structuration des données sous forme de types. Les spécificités du C sont abordées, notamment le passage de paramètres, l'utilisation des pointeurs, et l'allocation dynamique. La pile d'exécution qui met en œuvre les différents mécanismes d'allocation et d'appel de fonctions est aussi présentée. Les bonnes pratiques de programmation C sont illustrées sur plusieurs exemples. Enfin, nous montrons comment le C peut aussi supporter des éléments de la programmation objet, comme le polymorphisme au travers des pointeurs de fonction ou l'héritage. Un support de cours complet est mis à disposition des étudiants, ainsi que des fiches préparant chaque cours.

Objectifs

Savoir décomposer un problème en un ensemble de structures de données et de fonctions en langage C. Comprendre la programmation impérative bas-niveau, la raison d'être des pointeurs, et la gestion de la mémoire. Savoir développer un programme en langage C conséquent, savoir utiliser des bibliothèques C, savoir implémenter des algorithmes en langage C, avec les structures de données adéquates. Savoir découper un programme en modules.

Heures d'enseignement

CM	Cours Magistral	7,5h
TD	Travaux Dirigés	7,5h
TP	Travaux Pratiques	12h

Pré-requis obligatoires

Avoir des bases solides dans un langage de programmation impératif, avoir quelques notions de langage C

Plan du cours

CM : 0 - éléments de base (structures de contrôle), 1 - les fonctions en C, 2 - les tableaux en C, 3 - structures et pointeurs en C, 4 - mécanismes d'allocation mémoire, 5 - structures auto-référentes, pointeurs de fonction, préprocesseur, 6 – modules et compilation séparée, fichiers.

TD: Une approche par problème est privilégiée pour les séances de TD (et certaines séances de CM). Ces problèmes nécessitent l'acquisition progressive des connaissances décrites dans les CM. Le premier problème est relatif à l'analyse simple de texte, ce qui fait travailler les structures de contrôle, l'appel de fonctions, la définition de types structurés, ainsi que les chaînes de caractères. Le deuxième problème fait travailler plus l'algorithmique dans des tableaux bidimensionnels.

Cela fait travailler les structures, la façon dont le C compile le code, et comment optimiser le code. Le dernier problème fait de l'analyse complexe de texte, avec des statistiques sur les mots utilisés. Cela permet de travailler les pointeurs, l'allocation dynamique et la gestion mémoire, la pile d'exécution, ainsi que les façons en C pour représenter des structures de données génériques de type collection (ici tableau associatif par table de hachage).

TP:

Le premier TP est un warm-up pour pratiquer la syntaxe du C sur des exemples simples. On se sert de gnuplot pour faire des affichages graphiques. Ensuite les deux TP suivants sont consacrés au développement d'un jeu Tetris temps-réel graphique. Le deuxième TP ne s'intéresse qu'à la logique du jeu et l'affichage se fait dans un terminal texte. Il fait travailler la mise au point de structures de données et d'algorithmes. Le dernier TP ne s'occupe que de la partie graphique du Tetris. La bibliothèque GTK est utilisée dans ce but. Cela permet aux étudiants de découvrir la compilation séparée, le développement d'IHM, la programmation événementielle, et le temps-réel via le déclenchement de timer. Enfin, l'affichage graphique se fait grâce à Cairographics.

Infos pratiques

Contacts

Responsable du cours

Jean-Yves Ramel

☎ +33 4 79 75 88 88

✉ Jean-Yves.Ramel@univ-savoie.fr

Lieux

➤ Le Bourget-du-Lac (73)

Campus

➤ Le Bourget-du-Lac / campus Savoie Technolac