NFA037 - Programmation en langage C/C++

Mis à jour le 22-04-2024

Code : NFA037

Unité d'enseignement de type mixte

6 crédits

Volume horaire de référence (+/- 10%) : **50 heures**

Responsabilité nationale : EPN05 - Informatique / 1

Contact national :

EPN05 - Informatique

2 rue Conté
75003 Paris

Jean-mathieu Codassé
jean-mathieu.codasse@lecnam.net

Présentation

Prérequis

Publics visés

- Personnes souhaitant apprendre à bien programmer,
- Personnes souhaitant approfondir leur niveau de programmation avec les langages C et C++,
- Personnes souhaitant comprendre la programmation orientée objet avec C++,
- Personnes souhaitant intégrer des écoles d'informatique, de programmation type école 42,
- Développeurs, scientifiques, chefs de projet, enseignants, informaticiens, électroniciens, mathématiciens, ingénieurs, analystes financiers, comptables...
- Etudiants, apprentis, élèves ingénieurs, élèves d'écoles de commerce,
- Enseignants de la spécialité NSI,
- Candidats au CAPES d'informatique.

Prérequis
Il est conseillé d'avoir quelques rudiments de programmation car le rythme est soutenu.
Posséder des bases de programmation, quel que soit le langage (variables, affectations, tests conditionnels, boucles, tableaux, fonctions) est un atout sans être obligatoire.

Néanmoins, si vous travaillez régulièrement et respectez bien les conseils donnés vous pourrez suivre et valider cette UE.

Si vous voulez vraiment progresser : prévoir un minimum de 2 x 2 heures de programmation par semaine.

Avoir le niveau d'UE comme : NFE006 Macros/VBA, NFE003 (comprendre l'informatique et Python) , NFA031 (Java) est un atout mais n'est pas obligatoire.

Après cette UE...

Cette UE peut-être suivie d'unités d'enseignement en programmation du département d'informatique ou d'analyse numérique mathématique (CS001 et CSC002).

Vous pourrez accéder à des UE du département informatique sur la recherche opérationnelle, l'intelligence artificielle, la programmation embarquée...

Objectifs pédagogiques

Objectifs

- Apprendre à bien programmer,
- En validant cette UE on **obtient un très bon niveau d'autonomie de programmation** avec les langages C et C++, langages les plus utilisés dans le monde avec Java et Python,
- **Perfectionner** son niveau de programmation en C et C++,
- Comprendre et maîtriser les principes de la **programmation orientée objet** avec le langage C++,
- Interfaçage C++ / Python

Organisation de l'enseignement

Cet enseignement se veut concret, pragmatique avec une progression pédagogique étudiée et graduée ; un seul objectif : savoir programmer pleinement et en autonomie en C et C++.

Si l'UE est ouverte en **Présentiel** : une séance sur machine de 3,5 heures, chaque semaine pendant 13 semaines (Cours/ED/TP).

Si l'UE est ouverte en **Formation Ouverte à Distance (FOAD) :** documents pédagogiques, cours, exercices, corrigés et vidéos sont mis en ligne semaine après semaine permettant de travailler de chez soi, à son propre rythme ; des séances en visio permettent de faire le point avec l'enseignant, en direct ; un forum permet des échanges concrets et réguliers de questions et réponses...

Si l'UE est ouverte en "**Hybride" :** nombreux documents pédagogiques, cours, exercices, corrigés et vidéos mis en ligne chaque semaine + au moins 2 séances de regroupements sur place ou en visio pour faire le point, reprendre des éléments de cours, corriger des exercices, répondre aux questions, etc.

Programme

Contenu

Programme détaillé de l'UE

- Présentation, reprise des bases du langage C (variables, typage, entrées/sorties, tests conditionnels, opérateurs de pré et post incrémentation...),
- Rappels sur les boucles for et while, imbrications de boucles,
- Les tableaux, tableaux 2D, dynamiques, algorithmes,
- Les fonctions, passage de paramètres par valeur, valeurs de retour,prototypes...
- La récursivité, fonctions récursives,
- Les pointeurs, adresses, gestion de la mémoire, passage de paramètres par adresse, accès direct en mémoire
- Les types énumérés et complexes, créer son propre type de données,
- Gestion des chaînes de caractères, fonctions utiles sur les chaînes de caractères, algorithmes,
- Lecture et écriture dans des fichiers textes et binaires,
- Les listes chaînées, piles, files : ajout/suppression de maillons, parcours de listes, algorithmes avec des listes...
- De C à C++ : Écriture de programmes en C++,
- Implémentations en C++ d'arbres simples, parcours simples d'arbres, algorithmes récursifs,
- Introduction à la programmation orientée objet, notions de classes, méthodes, attributs,
- Les constructeurs, l'encapsulation de données, la surcharge d'opérateurs et de méthodes, destructeurs
- Setters, getters, méthodes statiques, dynamiques,
- L'héritage, le polymorphisme,
- L'héritage multiple, polymorphisme, masquage de fonction
- Généricité, templates, classes et méthodes virtuelles
- Gestion des exceptions
- Headers, programmation modulaire,
- Écriture et codage d'algorithmes, tris, tris récursifs...
- Comparaison de programmes C/C++ et Python
- Lancement de scripts Python à partir de C++
- Ecriture d'un script Python qui. lance un code C++

Programme non contractuel

Plus de 350 programmes concrets seront à écrire, programmer et compiler dans le cadre de cet enseignement.

Modalités de validation

- Contrôle continu
- Projet(s)

Description des modalités de validation

Projet individuel (code source C/C++) à rendre avec soutenance obligatoire et/ou examen individuel (code source C/C++ à écrire) dans le cadre du contrôle continu (pas de QCM).

Bibliographie

Titre	Auteur(s)
Documents de cours	xx
Exercices et corrigés d'exercices	xx
Vidéos pédagogiques	xx
Programmes sources implémentés	xx