

Programmation 2

| | |
|--------------------------|---|
| Domaine | Ingénierie et Architecture |
| Filière | Informatique et systèmes de communication |
| Orientation | Systèmes informatiques embarqués (ISCE) |
| Mode de formation | Temps partiel/En emploi |

Informations générales

| | |
|--------------------|------------------------|
| Nom | : Programmation 2 |
| Identifiant | : PRG2 |
| Années académiques | : 2021-2022, 2022-2023 |
| Responsable | : René Rentsch |
| Charge de travail | : 90 heures d'études |
| Périodes encadrées | : 64 (= 48 heures) |

| Semestre | E1 | S1 | S2 | E2 | S3 | S4 | E3 | S5 | S6 | E4 | S7 | S8 |
|-------------|----|----|----|----|----|----|----|----|----|----|----|----|
| Cours | | | 32 | | | | | | | | | |
| Laboratoire | | | 32 | | | | | | | | | |

Connaissances préalables recommandées

L'étudiant-e doit connaître et savoir utiliser les concepts de la programmation procédurale : types de base et types structurés, structures de contrôle, fonctions et unités de compilation.

L'unité PRG1 permet d'acquérir ces connaissances.

Objectifs

A l'issue de cette unité d'enseignement, l'étudiant-e sera capable de :

- appliquer tous les éléments du langage C étudiés (en particulier les pointeurs);
- analyser et implémenter proprement des applications de complexité moyenne;
- expliquer et justifier les choix réalisés, critiquer une solution proposée.

A l'issue des travaux pratiques en laboratoire, l'étudiant-e sera en outre capable de :

- utiliser les structures de données et de contrôle adaptées aux problèmes posés;
- analyser un problème et créer une application de complexité moyenne, en justifiant les choix réalisés;
- critiquer des solutions proposées;
- structurer correctement un code;
- rechercher et corriger des erreurs, adapter des solutions existantes;
- travailler seul ou de s'adapter dans un groupe.

Contenu et formes d'enseignement

Répartition des périodes indiquée à titre informatif.

Cours: 32 périodes

- 1. Introduction : Principales différences entre C et C++; préprocesseur; constantes symboliques, macros; compilation conditionnelle; types de base; opérateurs; entrées/sorties (scanf/printf) 6
- 2. Pointeurs : Déclaration d'un pointeur; arithmétique des pointeurs; tableaux et pointeurs; pointeur sur fonction; tableaux de pointeurs; allocation dynamique de la mémoire (alloc/malloc/realloc); restitution de la mémoire (free); fonctions prédéfinies de manipulation de la mémoire (memcpy, etc.) 10
- 3. Types composés : Struct; union; typedef; listes chaînées 5
- 4. Chaînes de caractères : Littéraux et variables; entrées/sorties; fonctions prédéfinies de manipulation (strcpy, strcat, etc); fonctions prédéfinies de conversion type numérique <-> chaîne de caractères; tableaux de chaînes de caractères; arguments de la ligne de commande 5
- 5. Fichiers : Lecture/écriture dans fichier texte; lecture/écriture dans fichier binaire; accès séquentiel vs accès direct 6

Laboratoire: 32 périodes

- Pointeurs, gestion dynamique de la mémoire 8
- Types composés (struct et unions) 8
- Chaînes de caractères 8
- Fichiers textes et binaires 8

Bibliographie

Le guide complet du langage C, Claude Delannoy, Eyrolles 2020

Contrôle de connaissances

Cours : l'acquisition des matières de cet enseignement sera contrôlée au fur et à mesure par des tests et des travaux personnels tout au long de son déroulement. Il y aura au moins 2 tests d'une durée totale d'au moins 2 périodes.

Laboratoire : ils seront évalués sur la base des rapports de manipulation, à 2 reprises au minimum.

Examen : L'atteinte de l'ensemble des objectifs de formation sera vérifiée lors d'un contrôle final commun écrit d'une durée de 90 minutes.

Matériel autorisé :

- Information communiquée directement par l'enseignant.

Calcul de la note finale

Note finale = moyenne cours x 0.3 + moyenne laboratoire x 0.2 + moyenne examen x 0.5