

Programmation 1

| | |
|--------------------------|---|
| Domaine | Ingénierie et Architecture |
| Filière | Informatique et systèmes de communication |
| Orientation | Sécurité informatique (ISCS) |
| Mode de formation | Temps partiel/En emploi |

Informations générales

| | |
|--------------------|-----------------------|
| Nom | : Programmation 1 |
| Identifiant | : PRG1 |
| Année académique | : 2025-2026 |
| Responsable | : Olivier Cuisenaire |
| Charge de travail | : 270 heures d'études |
| Périodes encadrées | : 192 (= 144 heures) |

| Semestre | E1 | S1 | S2 | E2 | S3 | S4 | E3 | S5 | S6 | E4 | S7 | S8 |
|-------------|----|----|----|----|----|----|----|----|----|----|----|----|
| Cours | | 96 | | | | | | | | | | |
| Laboratoire | | 96 | | | | | | | | | | |

Connaissances préalables recommandées

L'étudiant-e doit :

- savoir utiliser un système d'exploitation (Windows, Mac, Linux) ;
- posséder des connaissances de base en matière d'outils bureautiques ;
- savoir utiliser Internet ;
- posséder des notions de base de programmation.

Objectifs

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

- appliquer tous les aspects séquentiels du langage C++ et les bases de l'objet ;
- 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: 96 périodes

- 1. Introduction : Histoire du C++; Hello, World!; Compilation / Débogage / Exécution; Contrôle de version avec git. 6
- 2. Eléments de base : Variables, types de base; opérateurs, références et pointeurs, priorité des opérateurs, commentaires 6
- 3. Structures de contrôle : Notion de bloc; instruction de test et de boucles 6
- 4. Fonctions : Motivation; définition d'une fonction; modes de passage des paramètres; valeur de retour; variable locale vs globale à une fonction; appel d'une fonction; compilation séparée; compilation en ligne de commande; variable static et fonction static; valeur par défaut des paramètres 6
- 5. Flux : lecture et écriture à la console et dans des fichiers 6
- 6. Types arithmétiques : entiers signés et non signés, réels. Représentation en mémoire, limites, précision, affichage 6
- 7. Structures et énumérations 6
- 8. Chaînes de caractères : string et string_view 6
- 9. Tableaux : array, vector et span 6
- 10. Fonctions avancées : surcharge et généricité 6
- 11. Classes : fonctions membres; privé vs. publique, constructeurs et destructeurs, surcharge des opérateurs, attributs et fonctions membres statiques 6
- 12. Itérateurs, bibliothèques algorithm et numeric 6
- 13. Classes génériques 6
- 14. Gestion des erreurs : assertions, exceptions, terminaison du programme 6
- 15. Allocation dynamique : new, new[], delete, delete[], RAII 6
- Tests écrits 6

Laboratoire: 96 périodes

- Les laboratoires incluent des exercices courts disponibles via https://github.com/PRG1-HEIGVD/PRG1_Recueil_Exercices, et des exercices plus longs qui changent d'année en année et peuvent varier d'une classe à l'autre en adaptation aux besoins pédagogiques identifiés par le professeur 96

Bibliographie

1. Programmer en C++ moderne, Claude Delannoy, Eyrolles 2019
2. Big C++, Cay S. Horstmann, Wiley 2017

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 3 tests d'une durée totale d'au moins 6 périodes.

Laboratoire : ils seront évalués sur la base de rapports de manipulation ou de tests en classe, à 3 reprises minimum.

Examen : l'atteinte de l'ensemble des objectifs de formation sera vérifiée lors d'un contrôle final commun écrit d'une durée de 180 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