

# Programmation générique



## En bref

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

## Présentation

### Description

Ce module a un double objectif. D'une part il donne les éléments principaux de la programmation C++, en supposant connu le C : références, programmation objet classique avec héritage et polymorphisme, surcharge d'opérateurs, allocation dynamique. L'étonnante richesse et versatilité du langage est mise en évidence sur de nombreux exemples (programmation impérative, bas-niveau, objet, fonctionnelle). D'autre part il expose les intérêts et introduit les ingrédients principaux de la programmation générique: fonctions et classes patron, bibliothèque générique STL, concepts et modèles de concepts, classes traits et méta-programmation. Un support de cours complet est mis à disposition des étudiants.

### Objectifs

Connaître les principales caractéristiques du langage C++, et identifier quand certains paradigmes de programmation sont préférables à d'autres (e.g. polymorphisme dynamique vs généricité). Savoir programmer objet en C++, écrire des classes abstraites et utiliser le polymorphisme dynamique. Reconnaître quand utiliser la programmation générique et savoir la mettre en œuvre. Savoir identifier les abstractions dans les algorithmes et les structures de données, pour ensuite pouvoir mettre en œuvre un code générique adapté aux concepts identifiés.

---

## Heures d'enseignement

CM	Cours Magistral	9h
TD	Travaux Dirigés	9h
TP	Travaux Pratiques	12h

---

## Pré-requis obligatoires

Avoir des bases solides dans le langage C ainsi que dans un langage de programmation objet (e.g. JAVA).

---

## Plan du cours

- \* **CM** : Le cours présente d'abord l'intérêt et la puissance du langage C++, puis décrit les principaux éléments du langage C++ : les similitudes avec le C, les références, la programmation orientée objet, les hiérarchies de classes, les classes abstraites et le polymorphisme dynamique, l'utilisation des pointeurs dans ce paradigme de programmation. Ensuite la programmation générique est abordé, d'abord du point de vue des fonctions génériques ( patrons de fonction), puis du point de vue des structures de données ( patrons de classes). L'identification des concepts sous-jacents à une formulation générique est expliqué. Enfin, les autres ingrédients de la programmation générique sont abordés : bibliothèque standard C++ (notamment conteneurs, itérateurs, et algorithmes), spécialisations de classes, classes traits et méta-programmation.
- \* **TD** : Chacun des TDs permet de mettre en pratique les différents concepts vus en cours. Les premiers TDs font travailler le C++ "classique", notamment les références et le polymorphisme. L'exemple d'un ensemble de formes avec des algorithmes aléatoires "presque sûrs" de tests d'inclusion permet de faire travailler la programmation orientée objet et la syntaxe C++. Ensuite, la programmation générique est travaillée à la fois sous ses aspects fonctions et classes. On montre l'intérêt de la spécialisation des conteneurs sur quelques exemples concrets, comme les "smart pointers".
- \* **TP** : Un premier TP permet de se familiariser avec la programmation objet classique en C++, et illustre l'intérêt du polymorphisme dynamique pour manipuler des collections hétéroclites d'objets. On fournit une application graphique Qt qui gère des objets graphiques en mouvement. L'étudiant enrichit les objets graphiques, leurs formes, leurs mouvements, et met en place des algorithmes pour détecter les collisions. Ensuite, 2 TPs utilisent un même fil rouge, le traitement d'image, pour introduire la programmation générique. D'abord on met en œuvre une programmation orientée objet classique C++, en définissant une classe image, des opérateurs d'import/export (sérialisation), des traitements simples (seuillage, histogramme, égalisation). L'étudiante pratique aussi un peu la bibliothèque STL. Ensuite on aborde la programmation générique proprement dite. Le conteneur Image devient générique et peut traiter de façon indifférente images couleur ou noir et blanc. Des itérateurs génériques et des adaptateurs sont mis au point pour traiter les images de couleurs, afin de pouvoir accéder à leurs canaux couleur ou leur intensité comme si c'était des images en niveaux de gris. Certains algorithmes de filtrage ou de rehaussement fonctionnent alors sur des images arbitraires.

## Infos pratiques

## Lieux

- › Le Bourget-du-Lac (73)
- 

## Campus

- › Le Bourget-du-Lac / campus Savoie Technolac