Introduction

Remerciements

Puisqu'il faut bien commencer quelque part, je vais commencer par la partie que les lecteurs survolent généralement. Bien que ce livre est le fruit du travail d'un seul auteur, beaucoup de personnes ont contribué, directement ou indirectement, à ce cours.

En premier lieu, il faut remercier les personnes qui ont pris le temps de relire, de corriger et de proposer des remarques pour améliorer ce cours.

Je dois remercier aussi tous ceux qui partagent leurs connaissances sur les forums C++, ceux avec qui j'ai pu avoir de nombreuses discussions sur les problématiques spécifiques du C++, sur la conception et les cycles de vie des logiciels en général, sur la pédagogie pour enseigner la programmation.

Il me faut remercier aussi tous les débutants motivés, qui améliorent les choses par leurs questions. Le fait d'en voir certains qui arrivent débutants sur les forums C++ et qui, quelques mois plus tard, montrent qu'ils ont assimilé les notions que l'on a essayé de transmettre et qui aident à leur tour les autres débutants est une source de motivation importante pour continuer à prendre du temps pour rédiger des cours, articles ou réponses détaillées aux questions des débutants.

Et bien sur, je ne peux m'empêcher de remercier tous les enseignants qui n'ont pas su suivre l'évolution du C++, de sa philosophie ou du génie logiciel en général...

De quoi va parler ce cours ?

Donner un titre à un livre est assez complexe. Il faut qu'il soit

suffisamment informatif, mais pas trop long pour être percutant. J'avais, dans un premier temps, choisi comme titre “Cours pratique de création d'application en C++ moderne”, ce qui me semblait être le plus descriptif possible. Malheureusement, ce titre est très long et je suis passé à “Cours de C++ moderne”.

Pourquoi j'avais choisi ce premier titre ? Voyons les termes utilisés, un par un :

Cours

À la différence d'un article ou d'un tutoriel, un cours est conçu selon une approche et un objectif pédagogique. Le but n'est pas d'être un “dictionnaire” exhaustif du langage C++, comme le livre de Bjarne Stroustrup (le créateur du langage C++) : The C++ Programming Language. Le but n'est pas non plus d'être une source de référence, lorsque vous avez un doute sur un syntaxe particulière, comme le site cppreference.com.

D'ailleurs, plus généralement, ce cours n'est pas destiné a être votre seule source d'apprentissage du C++. Il existe beaucoup d'ouvrages et de sites sur le C++, destinés aux débutants comme aux développeurs confirmés, allant des problématiques spécifiques du C++ jusqu'aux différents domaines d'application (calcul scientifique, jeux vidéos, interfaces graphiques, etc).

Je ne vais pas citer tous les ouvrages intéressants à lire, il y en a trop. Vous pouvez avoir un aperçu sur la page dédiée aux livres sur mon site. Un point important quand même : n'hésitez surtout pas à poser des questions sur les forums AVANT d’acheter un livre. Même si les “bons” livres sont nombreux, les “mauvais” sont encore plus nombreux (en particulier en français). Ce cours ne doit pas faire exception, n'hésitez pas à poser des questions sur celui-ci avant de le suivre.

Le but de ce cours est de vous donner les éléments d'information de base pour démarrer votre apprentissage du C++. Cela veut dire par exemple que vous n'allez pas voir toutes les syntaxes possibles, mais vous saurez que vous n'avez pas vu toutes les syntaxes possibles. Vous saurez qu'il existe d'autres façon de faire les choses, chaque approche ayant ses

avantages et défauts.

Pratique

Un langage de programmation ne s'apprend pas dans les livres ou dans un cours. La programmation s'apprend par la pratique. Vous trouverez de nombreux exercices dans ce cours, dans le but de vous permettre de pratiquer chaque notion que vous aurez apprise. Une notion qui n'est pas pratiquée est une notion qui n'est pas assimilée, lire ce cours sans pratiquer les exercices ne vous permettra pas d'apprendre correctement.

Ce cours contient deux types de mise en application : les exercices et les projets.

Les exercices sont des utilisations directes du chapitre que vous venez d'étudier. Ils consistent en des codes à corriger ou modifier pour obtenir le résultat attendu. Il ne faut généralement que modifier ou ajouter quelques lignes de code et ils peuvent être réalisés en quelques minutes chacun.

Les projets nécessitent plus de travail. Ce sont des codes que vous allez créer sur le long terme, que vous allez conserver et modifier plusieurs fois au cours de votre progression dans ce cours. Vous pouvez même travailler à plusieurs sur un projet. Le but est de vous apprendre des notions importantes, mais qui ne sont pas accessibles lorsque vous travaillez sur de petits projets “jetables” : tout ce qui fait qu'un programme est de qualité, d'un point de vue professionnel.

Création d'application

L'objectif de ce cours n'est pas de vous apprendre le C++. Plus précisément, l'objectif n'est pas de vous apprendre simplement la syntaxe d'un langage de programmation.

Le but de ce cours n'est pas non plus de vous apprendre toutes les syntaxes possibles, toutes les problématiques que l'on peut rencontrer dans le C++. D'abord, parce que je ne les connais pas toutes :). Et si quelqu'un se lançait dans un tel ouvrage, cela s'appellerait “l'encyclopédie du C++”, il contiendrait probablement plus de 10.000

pages et serait obsolète au bout de six mois.

La programmation d'une application nécessite d'autres connaissances que la syntaxe, comme le cycle de vie des applications, la conception, l'algorithmie, savoir tester son code. Et plus généralement, tout l'écosystème du C++ : les outils, les bibliothèques, les méthodes de développement logiciel.

Moderne

Le C++ est un langage en constante évolution. La norme actuelle date de 2014 et la prochaine évolution du C++ est déjà planifiée en 2017. De nombreux cours sur internet ou dans les formations se basent sur l'ancienne norme du C++, le C++03 qui date de 2003. De plus, ces cours font généralement l'impasse sur ce qui fait la qualité d'un code, en particulier la gestion des erreurs.

Lorsqu'un débutant écrit un code d'apprentissage de quelques dizaines de lignes de code, ne pas respecter les bonnes pratiques ne sera pas critique. Mais dans un projet réel, dans le monde professionnel, cela sera catastrophique. C'est l'une des raisons qui fait que le C++ a parfois la réputation d'être un langage complexe. Et de mauvaises habitudes acquises lors de l'apprentissage seront difficiles à oublier par la suite.

Sommaire principal	Chapitre suivant