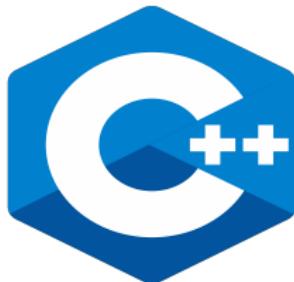


C++ : introduction

Achref El Mouelhi

Docteur de l'université d'Aix-Marseille
Chercheur en programmation par contrainte (IA)
Ingénieur en génie logiciel

elmouelhi.achref@gmail.com



1 Introduction

2 Installations

- Compilateur
- IDE

3 Avant de commencer

- Règles de nommage
- Instructions
- STL

4 Premier projet C++

C++

C++

- Langage de programmation
 - Compilé
 - Procédural et orienté-objet
 - Fortement typé : acceptant le typage statique, dynamique et générique
 - Sensible à la casse
- Créé par le professeur danois **Bjarne Stroustrup** dans les années 80 pendant qu'il travaillait pour les laboratoires **Bell** à New Jersey aux États-Unis
- Appelé initialement **C with Classes** en 1979 puis **C++** depuis 1983
- Extension (et évolution) du langage **C**
- Père des langages objets : **Java**, **C#**, **PHP**...
- Normalisé depuis 1998 par **ISO** (International Organization for Standardization)

C++

C++ : avantages

- C et C++ : langages les plus utilisés dans le monde
- Langage de bas niveau : rapide
- Disposant d'une bonne documentation, des supports vidéos, plusieurs exemples sur internet
- Énorme communauté : un des langages les plus utilisés dans le monde
- Portable : **Windows, Mac OS, Linux** (*Write once, run everywhere* ou Écrire une fois, exécuter partout)
- Multi-paradigme : possibilité de faire
 - la programmation procédurale
 - la programmation orientée objet (OOP)

C++ : inconvénient

complexe : gestion de mémoire (allocation dynamique, libération de la mémoire...)

C++ : quelques versions (normalisées avec ISO)

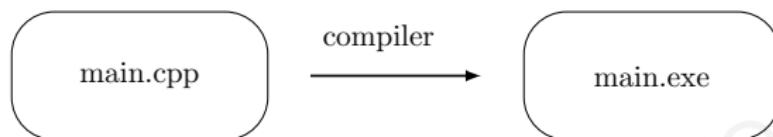
- **C++98** (sortie en 1998) : templates
- **C++03** (sortie en 2003)
- **C++11** (sortie en 2011) : mot-clé `auto`, expression Lambda...
- **C++14** (sortie en 2014) : fonctions Lambda génériques, attribut [deprecated] ...
- **C++17** (sortie en 2017) : quelques nouveaux attributs, initialisation dans `if` et `switch`...
- **C++20** (sortie en 2020) : opérateur de comparaison `<=>`...

Comment ça fonctionne ?

- On écrit un programme dans un fichier .cpp
- Ensuite, le compilateur génère un fichier .o du même nom (contenant du bytecode)
- Puis le linker charge les librairies référencées et génère un fichier .exe
- On lance le fichier .exe

main.cpp

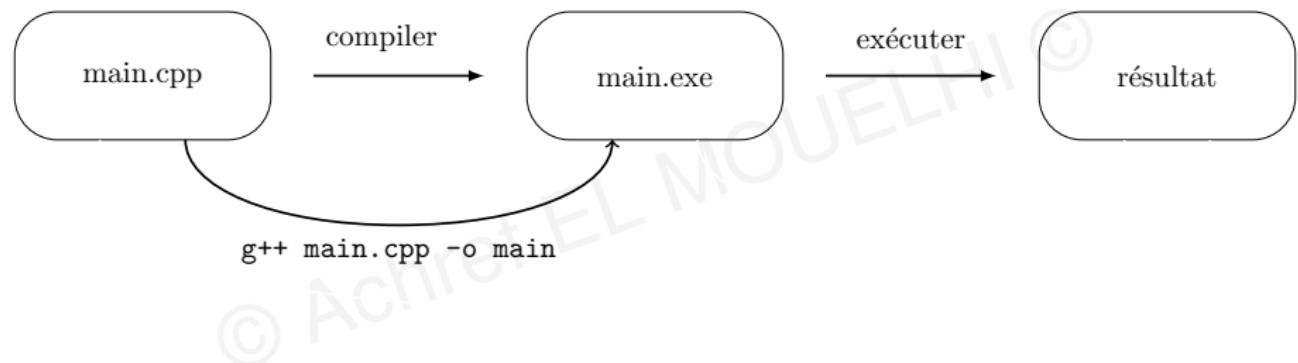
C++



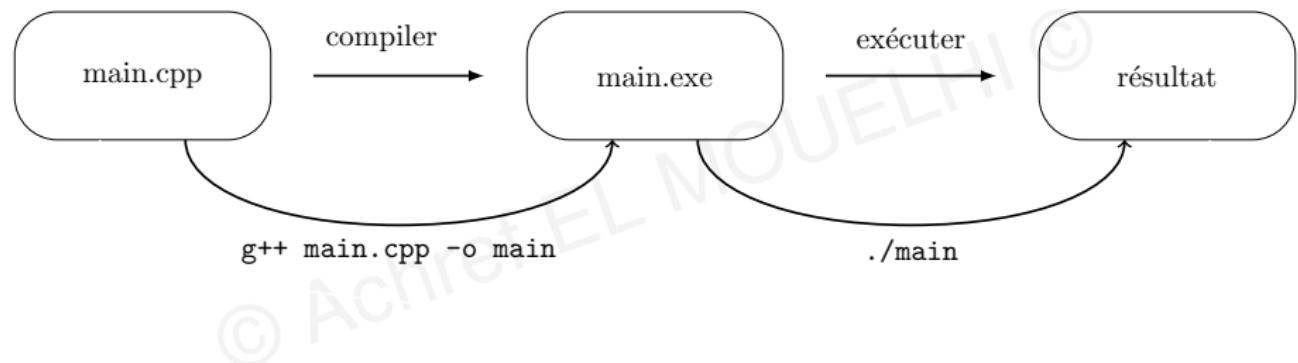
C++



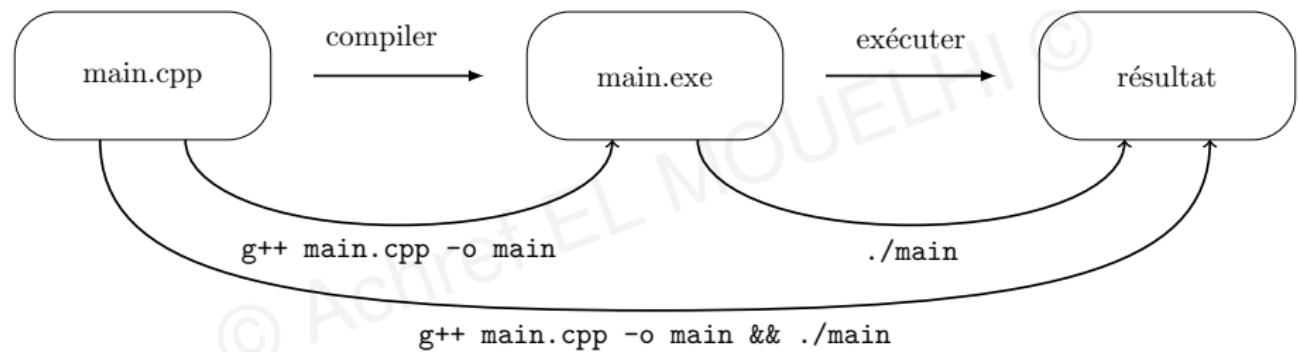
C++



C++



C++



De quoi on a besoin (le minimum) ?

- Un éditeur de texte (**Bloc-notes, Notepad++, Sublime Text, Atom...**)
- Un compilateur
 - **GCC (GNU Compiler Collection)** : une famille de compilateur pour les langages **C, C++, Java, Fortran...**, multi-plateforme (pour **Windows, Linux, Android**) et utilisé par le noyau du système **Unix**
 - `gcc` : pour **C**
 - `g++` : pour **C++**
 - `gcj` : pour **Java**
 - Clang : un compilateur pour les langages **C, C++ et Objective-C**, multi-plateforme (pour **Windows, Mac, Linux, Android, iOS**)
 - Sun Studio : un **IDE** (comprenant un compilateur) pour les langages **C, C++ et Fortran** développé par **Sun Microsystems**, multi-plateforme (pour **Linux, Solaris**)

Introduction

Pour installer **GCC (GNU Compiler Collection)**

- Aller à <https://www.msys2.org/>
- Suivre les différentes étapes d'installation
- Ajouter une variable de chemin pour le `bin` de **Mingw-w64** (défini généralement dans `C:\msys64`).

Introduction

Pour installer **GCC (GNU Compiler Collection)**

- Aller à <https://www.msys2.org/>
- Suivre les différentes étapes d'installation
- Ajouter une variable de chemin pour le `bin` de **Mingw-w64** (défini généralement dans `C:\msys64`).

Pour vérifier la version de `g++` installé, exécuter

```
g++ --version
```

C++

g++ version	C++ Standard
11	C++17
10	C++14
6	C++14
5	C++98

C++

On peut aussi utiliser un IDE (Environnement de développement intégré)

- pour éviter d'utiliser la console et les commandes
- pour avoir
 - l'auto-compilation
 - la coloration syntaxique
 - l'auto-complétion
 - l'indentation automatique...
 - ...
- pour une meilleure structuration du projet

C++

Exemple d'IDE pour C++

- **Visual Studio Code** : open-source, multi-plateforme (**Windows, Mac, Linux**)
- **CodeBlocks** : open-source, multi-plateforme (**Windows, Mac, Linux**) et écrit en **C++** utilisant par défaut GNU Gcc Compiler et supportant plusieurs autres
- **Visual C++** : disponible en version gratuite Visual C++ Express pour **Windows**
- **Eclipse** avec le plugin **CDT6** (open-source)
- **Netbeans** (open-source)
- **XCode** : pour **Mac**
- **DevC++** : pour **Windows**
- ...

C++

Visual Studio Code (ou VSC) , pourquoi ?

- Gratuit.
- Extensible selon le langage de programmation.

C++

Visual Studio Code (ou **VSC**) , pourquoi ?

- Gratuit.
- Extensible selon le langage de programmation.

VSC : téléchargement

code.visualstudio.com/download

Quelques raccourcis pour VSC

- Pour activer la sauvegarde automatique : aller dans File > AutoSave
- Pour indenter son code : `Alt` `Shift` `f`
- Pour commenter/décommenter : `ctrl` `:`
- Pour faire une sélection multiple : `Ctrl` `f2`
- Pour dupliquer une sélection : `Ctrl` `d`
- Pour consulter la liste des raccourcis : `Ctrl` `k` + `Ctrl` `s`

En cas de problème d'affichage de lettres accentués dans le terminal, exécutez

```
chcp 65001
```

Quelques extensions VSC pour C++

- **C/C++** : pour l'intelliSense
- **Code Runner** : pour l'ajout d'un bouton de compilation

Règles de nommage en C++

- Pour les classes et les fichiers : **Pascal case**
- Pour les variables, les objets, les fonctions et les méthodes : **Camel case**

Règles de nommage en C++

- Pour les classes et les fichiers : **Pascal case**
- Pour les variables, les objets, les fonctions et les méthodes : **Camel case**

Pour plus de détails

<https://wprock.fr/blog/conventions-nommage-programmation/>

C++

Instructions

- Chaque instruction se termine par ;
- Il est possible d'écrire plusieurs instructions sur une même ligne (**mais** ce n'est pas une bonne pratique)

STL : Standard Template Library

- { classes génériques }
- Fournissant des structures de données, des fonctions et des algorithmes
- Contenant 4 composants :
 - **Algorithms** : implémentation de certains algorithmes de recherche, de tri...
 - **Containers** : ensemble de structure de données comme liste, pile...
 - **Functions** : ensemble de fonctions génériques qui s'adaptent avec les différentes structure de données existantes
 - **Iterators** : objet permettant de parcourir une séquence de données

Démarche

- Créez un répertoire `CoursCpp` dans votre espace de travail
- Lancez **VSC** et allez dans `File > Open Folder...` et choisissez `CoursCpp`
- Dans `CoursCpp`, créez un fichier `main.cpp`

Ajoutons le code suivant dans `main.cpp`

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

C++

Explication

- `#include <iostream>` : la librairie permettant de gérer les flux entrant/sortant (input/output).
- `using namespace std` : un espace de nom est un genre de dossier abstrait (inexistant physiquement). Ici, on utilise l'espace de nom standard où toutes les librairies standards sont définies (**STL** : Standard Template Library).
- `int main()` : le point d'entrée vers notre projet (programme).
- `cout << msg <<` : permet d'afficher le message situé entre `<<` : cette opération est définie dans la librairie `<iostream>` de namespace `std`.
- `endl` : permet de forcer un retour à la ligne.
- `return 0` :
 - `main` doit retourner 0 pour une sortie sans erreur, autre valeur sinon.
 - Si aucun `return` n'est présent, le compilateur considère que la valeur de retour est par défaut 0.

Comment exécuter le programme ? (voir le résultat)

- Soit en faisant clic droit et ensuite choisir Run Code
- Soit en cliquant sur ▶ en haut à droite
- Soit avec le raccourci clavier **Ctrl** + **Alt** + **N**

C++

Comment exécuter le programme ? (voir le résultat)

- Soit en faisant clic droit et ensuite choisir Run Code
- Soit en cliquant sur ▶ en haut à droite
- Soit avec le raccourci clavier `Ctrl` + `Alt` + `N`

Constats

- Un fichier `main.cpp` généré.
- **Hello world !** s'affiche dans la console de **VSC**.