



## À la découverte du C++

PRUVOST Benjamin <sup>1</sup>

REPORT Tanguy <sup>2</sup>

1. [benjamin.pruvost@etu.univ-littoral.fr](mailto:benjamin.pruvost@etu.univ-littoral.fr)

2. [Tanguy.Report@etu.univ-littoral.fr](mailto:Tanguy.Report@etu.univ-littoral.fr)

# Table des matières

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>  | <b>2</b>  |
| 1.1      | L'histoire du C++ . . . . .                                | 2         |
| 1.2      | Différences entre C et C++ . . . . .                       | 3         |
| 1.3      | Installation de l'environnement de développement . . . . . | 3         |
| 1.3.1    | Windows . . . . .  | 3         |
| 1.3.2    | Linux . . . . .  | 4         |
| 1.3.3    | MacOS . . . . .  | 4         |
| <b>2</b> | <b>Structure d'un programme</b>                            | <b>5</b>  |
| <b>3</b> | <b>Types de données et variables</b>                       | <b>6</b>  |
| <b>4</b> | <b>Instructions de contrôle</b>                            | <b>7</b>  |
| <b>5</b> | <b>Fonctions</b>   | <b>8</b>  |
| <b>6</b> | <b>Pointeurs</b>   | <b>9</b>  |
| <b>7</b> | <b>Structures</b>  | <b>10</b> |

# Chapitre 1

## Introduction

### 1.1 L'histoire du C++



FIGURE 1.1 – Bjarne Stroustrup (2013)

Le langage C++ a été conçu au début des années 1980 par Bjarne Stroustrup (figure 1.1), un ingénieur d'origine danoise, alors qu'il travaillait au sein des laboratoires Bell Labs, un centre de recherche réputé pour ses innovations. Stroustrup souhaitait améliorer le langage C en y intégrant des fonctionnalités de programmation orientée objet, qui permettent de structurer le code de manière plus modulaire et intuitive, notamment grâce à des concepts tels que les classes et les objets. L'objectif principal était de combiner la puissance, la flexibilité et l'efficacité du C avec une organisation et une maintenabilité accrues, afin de faciliter l'écriture et la gestion de programmes complexes. Aujourd'hui, le C++ demeure l'un des langages de programmation les plus populaires et polyvalents, largement utilisé dans le développement de logiciels, de jeux vidéo, d'applications haute performance, ainsi que dans les systèmes embarqués et d'autres domaines technologiques avancés.

## 1.2 Différences entre C et C++

Le C et le C++ sont des langages de programmation étroitement liés, mais ils diffèrent par leurs caractéristiques et leurs usages. Le C est un langage procédural qui se concentre sur les fonctions et les structures de données, idéal pour les systèmes bas-niveau comme les systèmes d'exploitation. En revanche, le C++ étend le C en introduisant la programmation orientée objet, avec des concepts comme les classes, l'héritage et le polymorphisme, ce qui le rend plus adapté à des projets complexes nécessitant une organisation modulaire. De plus, le C++ offre des fonctionnalités modernes comme les templates, les exceptions et la gestion automatique des ressources, absentes en C. Malgré leurs différences, *le C++ reste compatible avec le C*, permettant d'utiliser du code C dans des projets C++.

## 1.3 Installation de l'environnement de développement

Afin d'écrire, compiler et exécuter du code C++ sur notre machine il est nécessaire d'installer deux outils essentiels : un éditeur de texte et un compilateur. Nous ne verrons pas l'installation de l'éditeur de texte dans ce cours il vous suffit de vous rendre sur le site de votre éditeur de texte de préférence, de télécharger le wizard d'installation correspondant à votre système et de suivre les indications pour pouvoir l'utiliser sur votre machine. Ces éditeurs incluent (mais ne se limitent pas à) :

- Visual Studio Code
- Notepad++
- Sublime Text

Concentrons nous donc sur l'installation du compilateur. Pour les utilisateurs de Mac, MacOS possède déjà un compilateur intégré, nous ne nous soucierons donc que des installations sur Windows et Linux.

### 1.3.1 Windows

1. Tout d'abord, rendez vous sur le site de MSYS2 et téléchargez l'installateur.
2. Lancez l'installateur et suivez les étapes.
3. Dans l'installateur, choisissez votre dossier d'installation, gardez le en mémoire pour plus tard. Dans la plus part des cas le dossier de base est acceptable.
4. Une fois l'installation terminée assurez vous que le choix *Run MSYS2 now* est coché puis sélectionnez *Finish*. Cela vous ouvrira un terminal MSYS2.
5. Dans ce terminal, installez les outils de compilation de MinGW en lançant la commande suivante :

```
pacman -S --needed base-devel mingw-w64-ucrt-x86_64-toolchain
```

6. Acceptez l'installation des packets en pressant la touche "Entrée".
7. Entrez 'Y' quand le programme vous demande si vous voulez poursuivre l'installation.
8. Ajoutez le chemin du dossier *bin* de votre installation de MinGW au PATH.
  - (a) Dans la barre de recherche Windows, cherchez *Modifier les variables d'environnement système*.
  - (b) Dans vos variables d'utilisateur, sélectionnez la variable *Path* et cliquez sur *Modifier*.

- (c) Sélectionnez *Nouveau* et ajoutez le chemin du dossier bin. Si vous n'avez pas changé l'adresse de celui proposé lors de l'installation le chemin devrait être le suivant :  
*C : \msys64 \ucrt64 \bin.*
  - (d) Sélectionnez *OK* deux fois.
9. Pour vérifier votre installation ouvrez un nouvel invite de commande puis tapez la commande suivante :

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

Si vous n'obtenez pas d'erreur alors félicitations ! Vous pouvez désormais compiler du code C++ !

### 1.3.2 Linux

Dans un premier temps, mettez à jour votre installateur en exécutant la commande suivante :

```
sudo apt update
```

Ensuite, installez l'outil de compilation *g++* grâce à cette autre commande :

```
sudo apt install g++
```

### 1.3.3 MacOS

Il est probable que le compilateur *g++* soit déjà installé sur votre système afin d'en être sûr vous pouvez exécuter cette commande et vérifier qu'elle ne vous renvoie pas d'erreur :

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

Dans le cas contraire vous pouvez l'installer grâce à la commande suivante :

```
xcode-select --install
```

## Chapitre 2

# Structure d'un programme

module de base (stdio.h), using namespace std, main()

## Chapitre 3

# Types de données et variables

types primitifs (int, float, char, bool) ;- tableau avec nom, encodage et range des valeurs.  
opérateurs simples

## Chapitre 4

# Instructions de contrôle

opérateurs de contrôle (`==`, `!=`, `<`, `>`), `if/else`, `while`, `for`, etc..



## Chapitre 5

# Fonctions

déclaration, passage de paramètres, retour de valeurs, (surcharge des fonctions ;- optionnel)

## Chapitre 6

# Pointeurs

opérations de base sur les pointeurs, explication en utilisant la mémoire, rapport avec les tableaux

## Chapitre 7

# Structures

introduction à l'OOP