

# Chapitre 1. Introduction

## Bibliographie

Sur papier :

- *Le langage C++*, Bjarne Stroustrup, Addison-Wesley, 4<sup>e</sup> édition, 2013
- *A Tour of C++*, Bjarne Stroustrup, Addison-Wesley, 2<sup>nde</sup> édition, 2018

En ligne :

Les bases (attention aux nouveautés, cours datés, mais bien faits) :

- <http://casteyde.christian.free.fr/cpp/cours/online/book1.html>
- [https://www.nawouak.net/informatics/french/index.htm#cpp\\_language](https://www.nawouak.net/informatics/french/index.htm#cpp_language) (Bruno Bachelet)

C++ avancé

- [http://cmm.ensmp.fr/~enficiau/Travaux/formation\\_cpp\\_transparents.pdf](http://cmm.ensmp.fr/~enficiau/Travaux/formation_cpp_transparents.pdf) (Raffi Enficiaud)
- Cours de 5A de M. Falcou
- <https://www.cppstories.com/>
- <https://www.stroustrup.com/>

C++ vs Java :

- [http://en.wikipedia.org/wiki/Comparison\\_of\\_Java\\_and\\_C%2B%2B](http://en.wikipedia.org/wiki/Comparison_of_Java_and_C%2B%2B) (à prendre avec des pincettes)

C++ étant un grand classique, les sources ne manquent pas : pensez à vérifier leur fiabilité et la date à laquelle elles ont été mises en ligne (question de norme et de version).

## Généralités sur la Programmation Orientée Objet

### La Programmation Orientée Objet

- Caractéristiques
  - Extensibilité
  - Réutilisabilité & Compatibilité
  - Validité et Robustesse

- Règle de fonctionnement

Choisissez vos classes. Fournissez un ensemble complet d'opérations pour chaque classe. Rendez toute similitude explicite à l'aide de l'héritage.

SLIDE 3

Coût des logiciels :

- 80% pour la maintenance dont 42% pour l'adaptation à de nouveaux besoins (extensibilité)

## Programmation Orientée Objet en C++

Les qualités requises pour un logiciel sont la validité, la robustesse, mais aussi l'extensibilité, la réutilisabilité et la compatibilité... la POO apporte ces deux dernières...

# Approche Objet : principe

NB : c'est une évolution de la programmation, ce n'est pas un idéal absolu.

Règle de fonctionnement : décomposer le modèle sous forme d'objets et d'interactions entre ces objets.

Difficultés principales :

- peu intuitif
- rien n'indique comment modéliser la structure d'un objet de façon pertinente
- il faut une grande rigueur
- PROGRAMMER EN C++ N'EST PAS CONCEVOIR OBJET

SLIDE 4

**Attention** : programmer en C++ n'est pas nécessairement concevoir objet, c'est l'analyse qui fait la conception objet.

## Histoire du langage C++

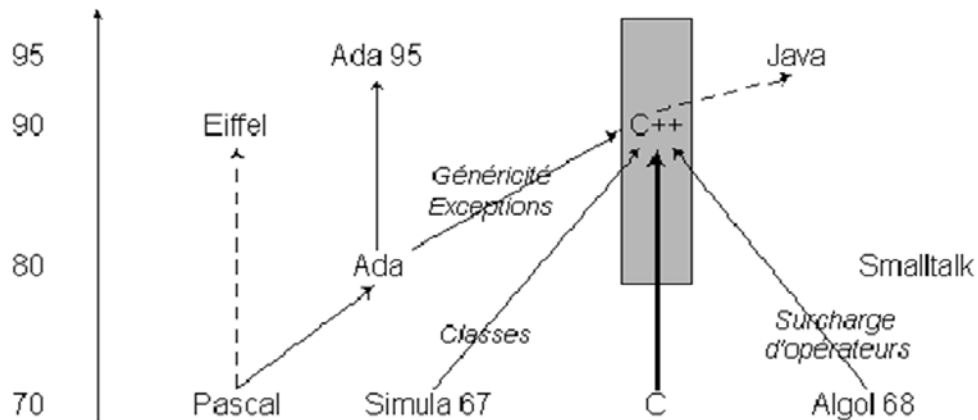
# Qu'est-ce que C++

- Inventé par Stroustrup à partir de :
  - C (Kernighan, 1978 – sous-ensemble)
  - Simula67 (Dahl, début 70 – notion de classe)
  - Algol68 (Woodward, 1974 – surcharge)
  - Depuis 1998, standard international (ISO/IEC 98-14882 et suivants. La dernière : ISO/CEI 14882:2011 )
- Caractéristiques :
  - Version améliorée de C
  - Supporte l'abstraction des données
  - Supporte la programmation OO
  - Supporte la programmation générique
  - Nouveautés compatibles Green Computing

SLIDE 5

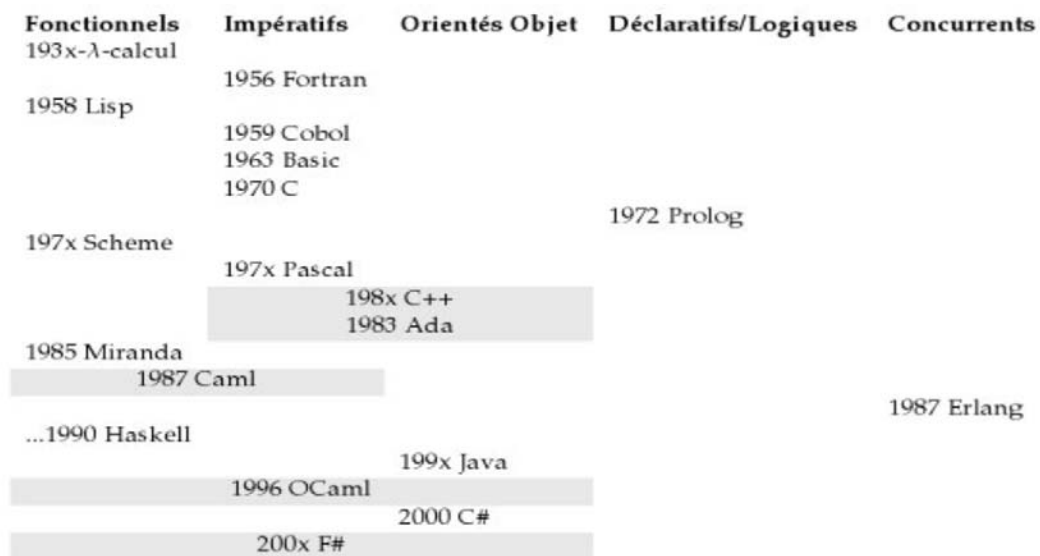
## Programmation Orientée Objet en C++

C++ a hérité des fonctionnalités de différents langages de programmation.



### SLIDE 6 : C++ par rapport aux autres langages

C'est un langage impératif ET orienté objet



Vincent Balat, <http://www.pps.jussieu.fr/~balat/pf/>

### SLIDE 7 : Types de programmation et langages

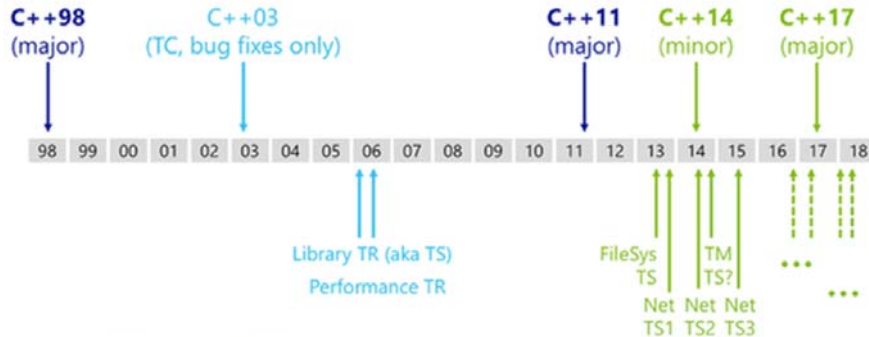
#### Définitions

La programmation impérative est un paradigme de programmation qui décrit les opérations en termes de séquences d'instructions exécutées par l'ordinateur pour modifier l'état du programme.

La programmation fonctionnelle est un paradigme de programmation qui considère le calcul en tant qu'évaluation de fonctions mathématiques et rejette le changement d'état et la mutation des données. Elle souligne l'application des fonctions, contrairement au modèle de programmation impérative qui met en avant les changements d'état.

## Programmation Orientée Objet en C++

NB : la plupart des langages de haut niveau comportent quatre types d'instructions principales : l'assignation ; le branchement conditionnel ; le branchement sans condition ; le bouclage.



© Sergio De Simone :

<http://www.infoq.com/news/2014/08/cpp14-here-features>

NB : finalement, C++ 17 est plutôt une évolution mineure ; la dernière évolution en date est de 2020.

### SLIDE 7 : C++ au fil du temps

La dernière norme ISO date de 2020 : C++ est un langage en évolution permanente : il existe toute une communauté qui développe et valide ce langage.

## Pourquoi ne pas enseigner seulement Java ?

### Pourquoi pas "juste java" ?

- C++ est...
  - Propre (enseignement)
  - Réaliste, efficace, souple (exigences)
  - Ouvert (collaboration entreprise)
  - Complet (concepts et techniques avancées)
  - Commercial
  - multiplateformes (mais attention compilateur)
- mais C++ n'est pas java...
  - Gestion de la mémoire
    - Pointeur, référence, valeur
    - Garbage collector / Ramasse-miettes (en Java... la politique C++ est de ne pas faire de miettes :-D)
  - Multi-plateforme
    - Machine virtuelle (Java)
    - Compilateur par architecture (C++)

C++ est beaucoup employé dans l'enseignement et la recherche car il est :

- suffisamment « propre » pour que l'enseignement des concepts de base de la programmation s'effectue dans de bonnes conditions
- suffisamment réaliste, efficace et souple pour satisfaire à l'élaboration de programmes exigeants

## Programmation Orientée Objet en C++

- suffisamment ouvert pour satisfaire les entreprises et les collaborations qui intègrent plusieurs environnements de développement et d'exécution
- suffisamment complet pour être un véhicule de l'enseignement des concepts et des techniques avancés
- suffisamment commercial pour que son utilisation survive largement à son enseignement dans le cadre académique