1. Introduction

Le langage C++, inventé par Bjarne Stroustrup vers 1983, est une évolution orientée objets du langage C de Brian Kernighan et Denis Ritchie.

Il s'est enrichi, au cours de la décennie 1980, parallèlement à la stabilisation et la normalisation de C (norme **C-Ansi** de 1989). C++ est actuellement en cours de normalisation.

2. Spécifications du langage

Le langage C++ est un sur-ensemble de C et repose sur les mêmes mécanismes d'écriture et de génération. De fait, une connaissance pratique de C est indispensable pour aborder le C++. Si C++ apporte, par rapport à C, des notions nouvelles et fondamentales, la syntaxe elle est très peu différente.
La syntaxe du C++ est 95 % de C et 5 % d'ajouts, et ce guide ne traite que des notions nouvelles (et des 5 % syntaxiques).

3. Extensions par rapport à C

3.1. Nouveaux concepts

C++ ajoute à C trois notions importantes et quelques améliorations, moins fondamentales mais intéressantes (Ces notions sont à développer prochainement) :

- **Classes et objets :** notion fondamentale et qui est le cœur même de la programmation orientée objets.
- **Surcharge de sélection :** c'est la possibilité de définir des traitements (fonctions) à un niveau conceptuel plus élevé qu'en programmation classique, en reportant sur le compilateur le maximum de détails de mise en œuvre, en particulier les traitements associés aux types des arguments d'appel.
- **Héritages :** c'est la possibilité de construire de nouveaux objets par réutilisation (dérivation) et modification d'objets existants (surcharge fonctionnelle).
- **Surcharges d'opérateurs :** c'est une extension des règles d'écritures arithmétiques classiques à des objets non scalaires.
- **Entrées/sorties par streams :** c'est une amélioration élégante des mécanismes de lecture/écriture de données.
- **Classe ou fonctions patrons (templates)** : Il s'agit d'un mécanisme d'aide à l'écriture des codes C++, permettant de définir, sous une forme semi-symbolique, des traitements similaires. Cette possibilité n'est disponible que depuis peu de temps dans les compilateurs C++.
- **Gestion d'erreurs et exceptions :** C++ a introduit, tout récemment, des principes sophistiqués de gestion des erreurs et problèmes en exécution.

3.2. Evolutions syntaxiques

Si la conception de programmes, en C++, n'a plus grand chose à voir avec la conception en C, la syntaxe elle est très peu modifiée par rapport au C-Ansi. Les différences portent sur :

- Introduction d'un commentaire "jusqu'à fin de ligne", symbolisé par // :

  ```
  int mini; // Valeur mini
  ```

  en plus du commentaire "bloc" de C /* ... */, toujours reconnu.
- Introduction de nouveaux mots-clés correspondant aux nouvelles fonctionnalités : class, public, private, protected, friend, virtual, this, template, operator, new, delete, et au support d'erreurs quand il est disponible : try, throw, rethrow, catch.
- Modification ou extension de la sémantique de certains mots-clés du C : extern, static, et de ertains opérateurs : &, >>, <<. C'est sur ces derniers points que les habitué(e)s de C devront se méfier.

4. Remarque :

Le choix de C++, pour une introduction à la conception objets, est dû au fait que c'est aujourd'hui un langage de développement orienté objets. Des langages antérieurs, comme SmallTalk, sont trop peu répandus pour justifier un investissement. D'autres, comme le récent langage Java, sont plus spécialisés et prennent donc tout leur intérêt dans des environnements bien particuliers (Internet).