PRG 1
en 15 chapitres

1. Introduction

- Une brève histoire du C++
- Hello world !
- Compiler, lier, déboger, exécuter
- L'Integrated Development Environment

2. Elements de base

- Variables et constantes
- Les types principaux : `int, double, char, bool, string, auto`
- Les opérateurs : `+, - , *, /, %, ++, --, <<, >>, …`
- Pointeurs et références

3. Structures de contrôle

- Branchements
  - `if … else`
  - `switch`
- Boucles
  - `while`
  - `for`
  - `do ... while`
- Sauts
  - `break; continue; goto`

4. Fonctions (notions de base)

- Structuration du code
- Passage des paramètres par
  - Valeur
  - Référence et référence constante
  - Adresse (pointeur)
- Récursivité
- Déclaration vs. Définition
- Compilation séparée

5. Flux

- Lecture / écriture
  - À la console
  - Dans un fichier
  - Dans une chaine de caractère
- Formatage de la sortie
- Gestion des erreurs d'entrée

6. Types arithmétiques et conversions

- Représentation en mémoire
- Valeurs représentables, comportement en cas de débordement
- Entiers
  - `short, int, long, long long, char`
- Entiers non signés
  - `unsigned short, unsigned int, unsigned long, unsigned long long, unsigned char, size_t`
- Réels
  - `float, double, long double`
- Conversions explicites et implicites

7. Structures et énumérations

- Créer de nouveaux types de données
  - En combinant plusieurs variables en une structure de données liées sémantiquement
  - En combinant plusieurs constantes correspondant à des alternatives

8. Chaines de caractère

- La classe `std::string`
- Accès aux caractères
- Concaténation
- Recherche
- Caractères et chaines étendues

9. Tableaux

- Stockage de nombreuses données de même type dans des emplacements mémoire consécutifs, accessibles via un indice
- `std::array`, tableaux de taille fixe, connue à la compilation
- `std::vector`, tableaux de taille variable
- `std::span`, interface commun à tous les tableaux

10. Fonctions surchargées et génériques

- Surcharge = fonctions de même nom mais différentiées par leurs paramètres
  - Quelle fonction est appelée parmi les surcharges ?
  - Comment éviter les ambiguïtés ?
- Généricité = surcharge avec des paramètres dont le type n'est pas spécifié a priori, mais fixé lors de l'appel

11. Classes

- Grouper ensemble les données d'une structure avec les fonctions qui les traitent.
- Encapsuler ces données et en restreindre l'accès direct
- Définir vos propres opérateurs

12. itérateurs, <algorithm> et <numeric>

- Traiter de manière uniforme tous les conteneurs de la STL
- Utiliser une boite à outil de traitements classiques sur des plages d'éléments stockés dans ces conteneurs

13. Classes génériques

- Créer des classes dont le type de données peut être spécifié par le code appelant

14. Gestion des erreurs

- Détecter et traiter les erreurs à la compilation, au débogage et à l'exécution avec
  - `static_assert`
  - `assert`
  - Le mécanisme d'exceptions

15. Allocation dynamique de la mémoire

- Tableaux classiques « à la C » et pointeurs
- Allocation dynamique des données et gestion des ressources RAII
- Swappable
- Movable