

Plan de cours de C++

David Odin

Septembre 2011

1 Lundi

1.1 Matin

- Introduction
- Contexte historique 1990, blabla startup
- Principales caractéristiques du langage C++ :
 - multi-paradigme
 - héritier du C, mais langage complètement séparé (méthodes de pensées)
 - La bibliothèque C est presque entièrement utilisable, `cstdio`.
 - réutilisabilité
- Chaîne de compilation, Makefile simple.
- **Syntaxe**
- Type de données
- **POD** :
 - `bool`, `char`, `int`, `float`, `double`, `void`
 - `long`, `short`, `unsigned`, `signed`
 - `const`, `volatile`
 - `enum`, `struct`, `class`, `typedef`, `union`
 - `static`, `extern`
 - `std` : `string`
- Données, constantes et expressions
- Variables
- Déclarations, définitions, expressions.
- **Exercices** déclaration / lecture de déclaration.

1.2 Après-midi

- Opérateurs, présentation. (support : liste des Opérateurs avec priorité : voir page 20)
- `sizeof`,
- Opérateurs de flux, `iostream` / `omanip` / `stringstream`
- **Exercices** : hello world, affichages formatés.
- **Exercices** utilisation de `cin` (cas simples)
- Contrôle d'exécution `while`, `do`, `for`, `continue`, `break`, `if`, `else`, `switch`, `case`, `default`, `goto`
- variables, portées, `shadow`

2 Mardi

2.1 Matin

- Fonctions, passage de paramètres et utilisation du code C dans un programme C++
- Pointeurs et références
- Références comme autre nom
- référence en retour de fonction

2.2 Après-midi

- polymorphisme fonctionnel
- Structures, union, enum
- protection des données
- amies
- this
- constructeurs, liste de construction, replicateur / destructeurs
- surcharge d'opérateurs
- opérateur (), fonctor
- opérateur de cast
- casting C++
- Membres et fonction membre (virtuelles / virtuelles pures)
- Dérivation et héritage (simple / multiple)
- class

3 Mercredi

3.1 Matin

STL

- Containers
- Iterators, reverse, copy, etc.
- Algorithmes (sort, utilisation de fonctor)

3.2 Après-midi

Conception Objet ?

Abstraction

Encapsulation

Modularité

- Espaces de noms

return,

delete, false, namespace, protected, template, try, virtual, catch, dynamic_cast, friend, new, public, this, typeid, wchar_t, class, explicit, inline, operator, reinterpret_cast, throw, typename, const_cast, export, mutable, private, static_cast, true, using,

Exercices de conception (héritage / composition)

4 Jeudi

4.1 Matin

Conception et organisation multi-fichier.

4.2 Après-midi

Aspects avancés - Modèles « templates » : définitions, syntaxe et instanciation

implémentation de `vec3<float,double,...>`

Retour sur la stl

- Approche de la gestion d'exceptions - Les exceptions dans C++ (notamment exceptions imbriquées) -
Les classes d'exception applicatives - Traitement des erreurs dans les programmes / en C++

5 Vendredi

- Design Patterns
 - Singleton
 - Factory
- Boost (signaux)
- C++-Ox : auto, forrange