



## C++ - Module 00

Namespace, class, member functions, stdio stream, initialization lists, static, const, et plein d'autres trucs basiques

*Résumé: Ce document contient le sujet du module 00 de la piscine C++ de 42*

# Chapitre I

## Règles Générales

- Toute fonction déclarée dans une header (sans pour les templates) ou tout header non-protégé, signifie 0 à l'exercice.
- Tout output doit être affiché sur stdout et terminé par une newline, sauf autre chose est précisé.
- Les noms de fichiers imposés doivent être suivis à la lettre, tout comme les noms de classe, les noms de fonction, et les noms de méthodes.
- Rappel : vous codez maintenant en C++, et plus en C. C'est pourquoi :
  - Les fonctions suivantes sont **INTERDITES**, et leur usage se soldera par un 0 : `*alloc`, `*printf` et `free`
  - Vous avez l'autorisation d'utiliser à peu près toute la librairie standard. CÉPENDANT, il serait intelligent d'essayer d'utiliser la version C++ de ce à quoi vous êtes habitués en C, plutôt que de vous reposer sur vos acquis. Et vous n'êtes pas autorisés à utiliser la STL jusqu'au moment où vous commencez à travailler dessus (module 08). Ça signifie pas de Vector/List/Map/etc... ou quoi que ce soit qui requiert une include `<algorithm>` jusque là.
- L'utilisation d'une fonction ou mécanique explicitement interdite sera sanctionnée par un 0
- Notez également que sauf si la consigne l'autorise, les mot-clés `using namespace` et `friend` sont interdits. Leur utilisation sera punie d'un 0.
- Les fichiers associés à une classe seront toujours nommés `ClassName.cpp` et `ClassName.hpp`, sauf si la consigne demande autre chose.
- Vous devez lire les exemples minutieusement. Ils peuvent contenir des prérequis qui ne sont pas précisés dans les consignes.
- Vous n'êtes pas autorisés à utiliser des librairies externes, incluant C++11, Boost, et tous les autres outils que votre ami super fort vous a recommandé
- Vous allez surement devoir rendre beaucoup de fichiers de classe, ce qui peut paraître répétitif jusqu'à ce que vous appreniez à scripter ça dans votre éditeur de code préféré.


- Lisez complètement chaque exercice avant de le commencer.
- Le compilateur est `clang++`
- Votre code sera compilé avec les flags `-Wall -Wextra -Werror`
- Chaque include doit pouvoir être incluse indépendamment des autres includes. Un include doit donc inclure toutes ses dépendances.
- Il n'y a pas de norme à respecter en C++. Vous pouvez utiliser le style que vous préférez. Cependant, un code illisible est un code que l'on ne peut pas noter.
- Important : vous ne serez pas noté par un programme (sauf si précisé dans le sujet). Cela signifie que vous avez un degré de liberté dans votre méthode de résolution des exercices.
- Faites attention aux contraintes, et ne soyez pas fainéant, vous pourriez manquer beaucoup de ce que les exercices ont à offrir
- Ce n'est pas un problème si vous avez des fichiers additionnels. Vous pouvez choisir de séparer votre code dans plus de fichiers que ce qui est demandé, tant qu'il n'y a pas de moulinette.
- Même si un sujet est court, cela vaut la peine de passer un peu de temps dessus afin d'être sûr que vous comprenez bien ce qui est attendu de vous, et que vous l'avez bien fait de la meilleure manière possible.

# Table des matières

<b>I</b>	<b>Règles Générales</b>	<b>1</b>
<b>II</b>	<b>Exercice 00 : Megaphone</b>	<b>4</b>
<b>III</b>	<b>Exercice 01 : My Awesome PhoneBook</b>	<b>5</b>
<b>IV</b>	<b>Exercice 02 : L'emploi de vos rêves</b>	<b>7</b>

# Chapitre II

## Exercice 00 : Megaphone


	Exercice : 00
Exercice 00 : Megaphone	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : <b>Makefile</b> , <b>megaphone.cpp</b>	
Fonctions interdites : <b>Aucune</b>	

Afin d'être sûr que tout le monde est bien reveillé, écrivez un programme dont le comportement est le suivant :

```
$>./megaphone "shhhhh... I think the students are asleep..."
SHHHHH... I THINK THE STUDENTS ARE ASLEEP...
$>./megaphone Damnit " ! " "Sorry students, I thought this thing was off."
DAMNIT ! SORRY STUDENTS, I THOUGHT THIS THING WAS OFF.
$>./megaphone
* LOUD AND UNBEARABLE FEEDBACK NOISE *
$>
```

# Chapitre III

## Exercice 01 : My Awesome PhoneBook

	Exercice : 01
Exercice 01 : My Awesome PhoneBook	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : <b>Makefile</b> , <b>*.cpp</b> , <b>*.{h, hpp}</b>	
Fonctions interdites : <b>Aucune</b>	

Bienvenue dans les années 80 et leur technologie incroyable! Écrivez un incroyable programme qui se comporte comme un annuaire **nul**.

Prenez le temps de donner à votre exécutable un nom pertinent. Lorsque le programme démarre, il demande l'input de l'utilisateur : vous devez accepter la commande **ADD**, la commande **SEARCH** ou la commande **EXIT**. Toute autre entrée est supprimée.

Le programme commence vide (pas de contacts), n'utilise pas d'allocation dynamique et ne peut pas stocker plus de 8 contacts. Si un neuvième contact est ajouté, veuillez définir un comportement pertinent.




<http://www.cplusplus.com/reference/string/string/> et bien sûr  
<http://www.cplusplus.com/reference/iostream/>

- Si la commande est **EXIT** :
  - Le programme se ferme et les contacts sont perdus à jamais
- Sinon, si la commande est **ADD** :
  - Le programme invitera l'utilisateur à saisir de nouvelles informations de contact, un champ à la fois, jusqu'à ce que chaque champ soit rempli.
  - Un contact contient les champs suivants : **first name**, **last name**, **nickname**, **phone number**, **darkest secret**.
  - Le PhoneBook doit être représenté comme une instance d'une classe dans votre code, il doit contenir un tableau de contacts.
  - Un contact **DOIT** être représenté comme une instance d'une classe dans votre code. Vous êtes libre de concevoir la classe comme vous le souhaitez, mais l'évaluation vérifiera la cohérence de vos choix. Regardez à nouveau les vidéos d'aujourd'hui si vous ne comprenez pas ce que je veux dire (et je ne veux pas dire "utilisez tout" avant de demander).
- Enfin, si la commande est **SEARCH** :
  - Le programme affichera une liste des contacts disponibles non vides dans 4 colonnes : **index**, **first name**, **last name** and **nickname**.
  - Chaque colonne doit avoir une largeur de 10 caractères, le contenu doit être aligné à droite, et chaque colonne séparée par le caractère ' | '. Toute sortie plus longue que la largeur des colonnes est tronquée et le dernier caractère affichable est remplacé par un point ( ' . ' ).
  - Ensuite, le programme demandera à nouveau pour l'index de l'entrée souhaitée et affichera les coordonnées du contact, un champ par ligne. Si l'entrée n'a aucun sens, définir un comportement pertinent.
- Sinon, l'input est ignoré.

Lorsqu'une commande a été correctement exécutée, le programme attend une nouvelle commande **ADD** ou **SEARCH** et il se termine si la commande **EXIT** est utilisée.

# Chapitre IV

## Exercice 02 : L'emploi de vos rêves

	Exercice : 02
Exercice 02 : L'emploi de vos rêves	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : <b>Makefile</b> , <b>Account.cpp</b> , <b>Account.hpp</b> , <b>tests.cpp</b>	
Fonctions interdites : <b>Aucune</b>	

C'est votre premier jour de travail à **GlobalBanksters United**. Vous avez réussi les tests de recrutement de l'équipe des programmeurs grâce à quelques astuces avec **Microsoft Office** qu'un ami vous a montrées. Mais vous savez que c'est votre installation rapide de **Adobe Reader** qui a époustoufflé votre recruteur. Cela vous a donné le petit avantage nécessaire pour battre vos adversaires pour ce poste.

Bref, vous avez réussi et votre patron vous a confié votre première tâche. Vous devez recoder un fichier source manquant car quelque chose a mal tourné. Il manque **Account.cpp**. Ils utilisent des clefs USB pour transférer les fichiers non git.

À ce stade, il sera légitime de quitter cet endroit, mais pour le bien de cet exercice, vous resterez.

Le fichier **Account.hpp** est présent et une compilation rapide de **tests.cpp** confirme qu'un **Account.cpp** est absent. Il y a aussi un ancien journal de sortie qui semble contenir la sortie correspondante.

Vous devez donc créer un fichier **Account.cpp** et écrire rapidement quelques lignes de pur génie en **C++**. Après quelques compilations ratées, votre programme se compilera et passera les tests avec une sortie parfaite, sauf pour les différences d'horodatage.

Bon sang, vous êtes bon !