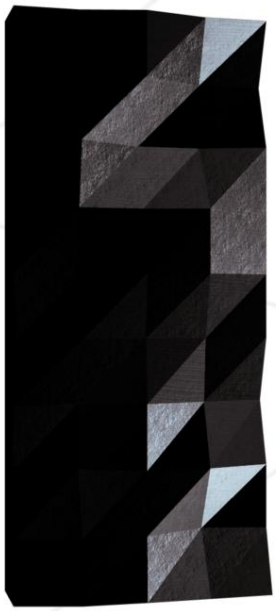

C++ - Module 00

Espaces de noms, classes, fonctions membres, flux stdio, listes d'initialisation, static, const et quelques autres éléments de base

Résumé :
Ce document contient les exercices du module 00 des modules C++.

Version : 7

Contenu

Chapitre I

Introduction

C++ est un langage de programmation à usage général créé par Bjarne Stroustrup comme une extension du langage de programmation C, ou « C avec classes » (source : Wikipédia).

L'objectif de ces modules est de vous initier à la programmation orientée objet.
Ce sera le point de départ de votre apprentissage du C++. De nombreux langages sont recommandés pour apprendre la POO. Nous avons choisi le C++, car il est dérivé du C.
Parce qu'il s'agit d'un langage complexe, et afin de garder les choses simples, votre code sera conforme à la norme C++98.

Nous sommes conscients que le C++ moderne est très différent à bien des égards. Si vous souhaitez devenir un développeur C++ compétent, il vous appartient de poursuivre vos études après le Common Core 42 !

Vous découvrirez de nouveaux concepts étape par étape. Les exercices gagneront progressivement en complexité.

Chapitre II

Règles générales

Compilation

- Compilez votre code avec c++ et les indicateurs -Wall -Wextra -Werror
- Votre code devrait toujours être compilé si vous ajoutez l'indicateur -std=c++98

Conventions de formatage et de dénomination

- Les répertoires d'exercices seront nommés de cette façon : ex00, ex01, ... , exn
- Nommez vos fichiers, classes, fonctions, fonctions membres et attributs comme requis dans les lignes directrices.
- Écrivez les noms de classe en majuscules . Les fichiers contenant le code de classe seront Il doit toujours être nommé selon le nom de la classe. Par exemple : ClassName.hpp/ClassName.h, ClassName.cpp ou ClassName.tpp. Ainsi, si vous disposez d'un fichier d'en-tête contenant la définition d'une classe « BrickWall » représentant un mur de briques, son nom sera BrickWall.hpp.
- Sauf indication contraire, chaque message de sortie doit être terminé par une nouvelle ligne caractère et affiché sur la sortie standard.
- Adieu Norminette ! Aucun style de codage n'est imposé dans les modules C++. Vous pouvez suivre votre style préféré. Mais gardez à l'esprit qu'un code que vos pairs ne comprennent pas est un code qu'ils ne peuvent pas noter. Faites de votre mieux pour écrire un code clair et lisible.

Autorisé/Interdit

Vous ne codez plus en C. Passez au C++ ! Par conséquent :

- Vous pouvez utiliser presque tout ce qui est proposé dans la bibliothèque standard. Ainsi, plutôt que de vous en tenir à ce que vous connaissez déjà, il serait judicieux d'utiliser autant que possible les versions C++ des fonctions C auxquelles vous êtes habitué.
- Cependant, vous ne pouvez utiliser aucune autre bibliothèque externe. Cela signifie que C++11 (et ses dérivés) et les bibliothèques Boost sont interdits. Les fonctions suivantes sont également interdites : *printf(), *alloc() et free(). Si vous les utilisez, votre note sera de 0, point final.

- Notez que, sauf indication contraire explicite, l'espace de noms d'utilisation <ns_name> et Les mots-clés amis sont interdits. Sinon, votre note sera de -42.
- Vous êtes autorisé à utiliser le STL uniquement dans le module 08. Cela signifie : pas de conteneurs (vecteur/ liste/carte, etc.) ni d'algorithmes (tout ce qui nécessite l'inclusion de l'en-tête <algorithm>) d'ici là. Sinon, votre note sera de -42.

Quelques exigences de conception

- Les fuites de mémoire se produisent également en C++. Lorsque vous allouez de la mémoire (en utilisant le nouveau (mot-clé), vous devez éviter les fuites de mémoire.
- Du module 02 au module 08, vos cours doivent être conçus dans le style orthodoxe Forme canonique, sauf mention explicite contraire.
- Toute implémentation de fonction placée dans un fichier d'en-tête (à l'exception des modèles de fonction) signifie 0 pour l'exercice.
- Vous devez pouvoir utiliser chacun de vos en-têtes indépendamment des autres. Ils doivent donc inclure toutes les dépendances nécessaires. Cependant, vous devez éviter le problème de double inclusion en ajoutant des gardes d'inclusion. Sinon, votre note sera de 0.

Lis-moi

- Vous pouvez ajouter des fichiers supplémentaires si nécessaire (par exemple, pour fractionner votre code). Ces devoirs n'étant pas vérifiés par un programme, n'hésitez pas à le faire, à condition de fournir les fichiers obligatoires.
- Parfois, les lignes directrices d'un exercice semblent courtes, mais les exemples peuvent montrer exigences qui ne sont pas explicitement écrites dans les instructions.
- Lisez chaque module en entier avant de commencer ! Vraiment, foncez.
- Par Odin, par Thor ! Utilisez votre cerveau !!!

Vous devrez implémenter de nombreuses classes. Cela peut paraître fastidieux,
à moins que vous ne soyez capable de créer un script dans votre éditeur de texte préféré.

Vous disposez d'une certaine liberté pour réaliser les exercices.
Cependant, suivez les règles obligatoires et ne soyez pas paresseux. Vous
Vous manquez beaucoup d'informations utiles ! N'hésitez pas à lire
concepts théoriques.

Chapitre III

Exercice 00 : Mégaphone

	Exercice : 00
Mégaphone	
Répertoire de remise : ex00/	
Fichiers à rendre : Makefile, megaphone.cpp Fonctions	
interdites : Aucune	

Juste pour vous assurer que tout le monde est réveillé, écrivez un programme qui se comporte comme suit :

```
$>./megaphone "chut... Je crois que les étudiants dorment..."
CHUT... JE CROIS QUE LES ÉTUDIANTS DORMENT...
$>./megaphone Zut Zut ! DÉSOLÉ " ! "Désolé les étudiants, je pensais que ce truc n'allait pas."
LES ÉTUDIANTS, JE PENSAIS QUE CE TRUC N'ÉTAIT PAS BON. $>./megaphone

* BRUIT DE RETOUR FORT ET INSUPPORTABLE * $>
```

Résolvez les exercices de manière C++.

Chapitre IV

Exercice 01 : Mon génial

Annuaire téléphonique

	Exercice : 01
Mon super annuaire téléphonique	
Répertoire de remise : ex01/	
Fichiers à remettre : Makefile, *.cpp, *.{h, hpp}	
Fonctions interdites : Aucune	

Bienvenue dans les années 80 et leur incroyable technologie ! Écrivez un programme qui se comporte comme un logiciel d'annuaire téléphonique génial et de mauvaise qualité.

Vous devez implémenter deux classes :

• Annuaire téléphonique

◦ Il dispose d'un ensemble de contacts.

◦ La mémoire peut stocker jusqu'à 8 contacts. Si l'utilisateur souhaite ajouter un neuvième contact, le nouveau remplace le plus ancien.

◦ Veuillez noter que l'allocation dynamique est interdite.

• Contact

◦ Représente un contact du répertoire téléphonique.

Dans votre code, le répertoire doit être instancié comme une instance de la classe PhoneBook . Il en va de même pour les contacts. Chacun d'eux doit être instancié comme une instance de la classe Contact . Vous êtes libre de concevoir les classes comme vous le souhaitez, mais gardez à l'esprit que tout ce qui est utilisé à l'intérieur d'une classe est privé, et que tout ce qui peut être utilisé en dehors d'une classe est public.

N'oubliez pas de regarder les vidéos intranet.

Au démarrage du programme, le répertoire est vide et l'utilisateur est invité à en saisir un de trois commandes. Le programme n'accepte que les commandes ADD, SEARCH et EXIT.

• AJOUTER : enregistrer un nouveau contact

◦ Si l'utilisateur saisit cette commande, il est invité à saisir les informations du nouveau contact, champ par champ. Une fois tous les champs complétés, le contact est ajouté au répertoire.

◦ Les champs de contact sont : prénom, nom, surnom, numéro de téléphone et Le secret le plus sombre. Un contact enregistré ne peut contenir aucun champ vide.

• RECHERCHE : afficher un contact spécifique

◦ Afficher les contacts enregistrés sous forme de liste de 4 colonnes : index, prénom, nom nom et surnom.

◦ Chaque colonne doit comporter 10 caractères de large. Elles sont séparées par une barre verticale (« | »). Le texte doit être aligné à droite. Si le texte est plus long que la colonne, il doit être tronqué et le dernier caractère affichable doit être remplacé par un point (« . »).

◦ Ensuite, demandez à nouveau à l'utilisateur l'index de l'entrée à afficher. Si l'index est hors limites ou incorrect, définissez un comportement approprié. Sinon, affichez les coordonnées, un champ par ligne.

• SORTIE

◦ Le programme se ferme et les contacts sont perdus à jamais !

• Toute autre entrée est rejetée.

Une fois qu'une commande a été correctement exécutée, le programme attend une autre. Il s'arrête lorsque l'utilisateur saisit EXIT.

Donnez un nom pertinent à votre exécutable.

http://www.cplusplus.com/reference/string/string/ et bien sûr http://www.cplusplus.com/reference/iomanip/

Chapitre V

Exercice 02 : Le travail de votre Rêves

	Exercice : 02
L'emploi de vos rêves	
Répertoire de remise : ex02/	
Fichiers à rendre : Makefile, Account.cpp, Account.hpp, tests.cpp Fonctions interdites :	
Aucune	

Account.hpp, tests.cpp et le fichier journal sont disponibles en téléchargement sur la page intranet du module.

Aujourd'hui, c'est votre premier jour chez GlobalBanksters United. Après avoir réussi les tests de recrutement (grâce à quelques astuces Microsoft Office qu'un ami vous a montrées), vous avez rejoint l'équipe de développement. Vous savez aussi que le recruteur a été impressionné par la rapidité avec laquelle vous avez installé Adobe Reader. Ce petit plus a fait toute la différence et vous a permis de vaincre tous vos adversaires (c'est-à-dire les autres candidats) : vous avez réussi !

Bref, votre responsable vient de vous confier du travail. Votre première tâche consiste à recréer un fichier perdu. Un problème est survenu et un fichier source a été supprimé par erreur. Malheureusement, vos collègues ne connaissent pas Git et utilisent des clés USB pour partager du code. À ce stade, il serait logique de quitter cet endroit immédiatement. Pourtant, vous décidez de rester. Défi relevé !

Vos collègues développeurs vous ont fourni un ensemble de fichiers. La compilation de tests.cpp révèle que le fichier manquant est Account.cpp. Heureusement, le fichier d'en-tête Account.hpp a été sauvegardé. Il existe également un fichier journal. Vous pourriez l'utiliser pour comprendre comment la classe Account a été implémentée.

Vous commencez à recréer le fichier Account.cpp. En quelques minutes seulement, vous codez quelques lignes de C++ de haute qualité. Après quelques compilations ratées, votre programme passe les tests avec succès. Son résultat correspond parfaitement à celui enregistré dans le fichier journal (à l'exception des horodatages qui seront évidemment différents, car les tests enregistrés dans le fichier journal ont été exécutés avant votre embauche).

Putain, tu es impressionnant !

Vous pouvez réussir ce module sans faire l'exercice 02.