



C++ - Module 07

Modèles C++

Résumé :

Ce document contient les exercices du module 07 des modules C++.

Version : 10.0

Contenu

I	Introduction	2
II	Règles générales	3
III	Instructions d'IA	6
IV	Exercice 00 : Commencer avec quelques fonctions	8
V	Exercice 01 : Iter	10
VI	Exercice 02 : Tableau	11
VII	Soumission et évaluation par les pairs	12

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 !

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 (UpperCamelCase) . Les fichiers contenant du code de classe seront toujours nommés selon le nom de la classe. Par exemple : `ClassName.hpp/ClassName.h`, `ClassName.cpp` ou `ClassName.tpp`. Ainsi, si vous avez 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 se terminer par un caractère de nouvelle ligne et être 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 que le code que vos pairs ne comprennent pas est du 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 les modules 08 et 09. Cela signifie : pas de conteneurs (vecteur/liste/carte, etc.) ni d'algorithmes (tout ce qui nécessite l'inclusion de l'en-tête <algorithm>) jusqu'à cette date. 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 09, vos cours doivent être conçus dans le style orthodoxe Forme canonique, sauf indication contraire explicite.
- 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 !!!



Concernant le Makefile pour les projets C++, les mêmes règles qu'en C s'appliquent (voir le chapitre Norme sur le Makefile).



Vous devrez implémenter de nombreuses classes. Cela peut paraître fastidieux, à moins que vous ne puissiez écrire des scripts 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

manquez beaucoup d'informations utiles ! N'hésitez pas à lire

concepts théoriques.

Chapitre III

Instructions de l'IA

- Contexte

Ce projet est conçu pour vous aider à découvrir les éléments fondamentaux de votre formation en TIC.

Pour ancrer correctement les connaissances et les compétences clés, il est essentiel d'adopter une approche réfléchie de l'utilisation des outils et du support de l'IA.

Un véritable apprentissage fondamental nécessite un véritable effort intellectuel — par le biais de défis, de répétitions et d'échanges d'apprentissage entre pairs.

Pour un aperçu plus complet de notre position sur l'IA — en tant qu'outil d'apprentissage, en tant qu'élément du programme des TIC et en tant qu'attente sur le marché du travail — veuillez vous référer à la FAQ dédiée sur l'intranet.

- Message principal

Construisez des bases solides sans raccourcis.

Développez réellement vos compétences techniques et énergétiques.

Vivez un véritable apprentissage entre pairs, commencez à apprendre à apprendre et à résoudre de nouveaux problèmes.

Le parcours d'apprentissage est plus important que le résultat.

Apprenez-en davantage sur les risques associés à l'IA et développez des pratiques de contrôle efficaces et des contre-mesures pour éviter les pièges courants.

- Règles de l'apprenant :

- Vous devez appliquer le raisonnement aux tâches qui vous sont assignées, en particulier avant de vous tourner vers l'IA.

- Vous ne devez pas demander de réponses directes à l'IA.
- Vous devriez en apprendre davantage sur l'approche globale 42 de l'IA.

• Résultats de la phase :

Au cours de cette phase fondamentale, vous obtiendrez les résultats suivants :

- Obtenez des bases techniques et de codage appropriées.
- Sachez pourquoi et comment l'IA peut être dangereuse durant cette phase.

• Commentaires et exemple :

- Oui, nous savons que l'IA existe et qu'elle peut résoudre vos projets. Mais vous êtes ici pour apprendre, pas pour prouver que l'IA a appris. Ne perdez pas votre temps (ni le nôtre) à simplement démontrer que l'IA peut résoudre un problème donné.

Apprendre à 42 ans ne consiste pas à connaître la réponse, mais à développer la capacité d'en trouver une. L'IA vous donne directement la réponse, mais cela vous empêche de construire votre propre raisonnement. Et le raisonnement demande du temps, des efforts et implique des échecs. Le chemin vers le succès n'est pas censé être facile.

- Gardez à l'esprit que pendant les examens, l'IA n'est pas disponible : pas d'Internet, pas de smartphones, etc. Vous vous rendrez vite compte si vous vous êtes trop appuyé sur l'IA dans votre processus d'apprentissage.
- L'apprentissage entre pairs vous expose à des idées et des approches différentes, améliorant ainsi vos compétences interpersonnelles et votre capacité à penser différemment. C'est bien plus précieux que de simplement discuter avec un robot. Alors n'hésitez pas : discutez, posez des questions et apprenez ensemble !
- Oui, l'IA sera intégrée au programme, à la fois comme outil d'apprentissage et comme sujet à part entière. Vous aurez même la possibilité de créer votre propre logiciel d'IA. Pour en savoir plus sur notre approche crescendo, vous consulterez la documentation disponible sur l'intranet.

Bonnes pratiques :


Je suis bloqué sur un nouveau concept. Je demande à quelqu'un à proximité comment il l'a abordé. On discute 10 minutes, et soudain, ça fait tilt. Je comprends.

Mauvaise pratique :

J'utilise l'IA en secret, je copie du code qui semble correct. Lors de l'évaluation par les pairs, je ne peux rien expliquer. J'échoue. À l'examen, sans IA, je suis à nouveau bloqué. J'échoue.

Chapitre IV

Exercice 00 : Commencez avec quelques fonctions

	Exercice : 00
Commencez avec quelques fonctions	
Répertoire de remise :	
ex00/ Fichiers à remettre : Makefile, main.cpp, whatever.{h, hpp}	
Fonctions interdites : Aucune	

Implémentez les modèles de fonctions suivants :

- swap : échange les valeurs de deux paramètres donnés. Ne renvoie rien.
- min : Compare les deux valeurs passées en paramètres et renvoie la plus petite.
S'ils sont égaux, il renvoie le deuxième.
- max : compare les deux valeurs passées en paramètres et renvoie la plus grande.
S'ils sont égaux, il renvoie le deuxième.

Ces fonctions peuvent être appelées avec n'importe quel type d'argument. La seule exigence est que les deux arguments soient du même type et prennent en charge tous les opérateurs de comparaison.



Les modèles doivent être définis dans les fichiers d'en-tête.

Exécution du code suivant :


```
int principal( vide ) {  
  
    int a = 2;  
    int b = 3;  
  
    ::échanger( a, b );  
    std::cout << "a = << a << ", b = << b << std::endl; "  
    std::cout << "min( a, b ) = << ::min( a, b ) << std::endl;  
    std::cout << "max( a, b ) = << ::max( a, b ) << std::endl;  
  
    std::string c = "chaîne1";  
    std::string d = "chaîne2";  
  
    ::échanger(c, d);  
    std::cout << "c = << c << ", d = << d << std::endl; "  
    std::cout << "min( c, d ) = << ::min( c, d ) << std::endl;  
    std::cout << "max( c, d ) = << ::max( c, d ) << std::endl;  
  
    retourner 0;  
}
```

Devrait afficher :

```
a = 3, b = 2  
min(a, b) = 2  
max(a, b) = 3  
c = chaîne2, d = chaîne1  
min(c, d) = chaîne1  
max(c, d) = chaîne2
```

Chapitre V

Exercice 01 : Iter

	Exercice : 01
Iter	
Répertoire de remise :	
ex01/ Fichiers à remettre : Makefile, main.cpp, iter.{h, cpp}	
Fonctions interdites : Aucune	

Implémentez un modèle de fonction iter qui prend 3 paramètres et ne renvoie rien.

- Le premier paramètre est l'adresse d'un tableau.
- Le deuxième est la longueur du tableau.
- La troisième est une fonction qui sera appelée sur chaque élément du tableau.

Soumettez un fichier main.cpp contenant vos tests. Fournissez suffisamment de code pour générer un exécutable de test.

Votre modèle de fonction itérative doit fonctionner avec tout type de tableau. Le troisième paramètre peut être un modèle de fonction instancié.


La fonction passée en troisième paramètre peut prendre son argument par référence constante ou par référence non constante, selon le contexte.



Réfléchissez bien à la manière de prendre en charge les éléments const et non const dans votre fonction iter.

Chapitre VI

Exercice 02 : Tableau

	Exercice : 02
Tableau	
Répertoire de remise : ex02/	
Fichiers à remettre : Makefile, main.cpp, Array.{h, hpp} et fichier optionnel : Array.tpp Fonctions interdites : Aucune	

Développer un modèle de classe Array qui contient des éléments de type T et qui implémente le comportement et les fonctions suivants :

- Construction sans paramètre : Crée un tableau vide.
- Construction avec un int non signé n comme paramètre : Crée un tableau de n éléments initialisé par défaut.
Astuce : essayez de compiler `int * a = new int();` puis affichez `*a`.
- Construction par opérateur de copie et d'affectation. Dans les deux cas, la modification de le tableau d'origine ou sa copie après copie ne doit pas affecter l'autre tableau.
- Vous DEVEZ utiliser l'opérateur `new[]` pour allouer de la mémoire. L'allocation préventive (allocation de mémoire à l'avance) est interdite. Votre programme ne doit jamais accéder à la mémoire non allouée.
- Les éléments sont accessibles via l'opérateur d'indice : `[]`.
- Lors de l'accès à un élément avec l'opérateur `[]`, si son index est hors limites, une `std::exception` est levée.
- Une fonction membre `size()` qui renvoie le nombre d'éléments du tableau. Cette fonction membre ne prend aucun paramètre et ne doit pas modifier l'instance courante.

Comme d'habitude, assurez-vous que tout fonctionne comme prévu et remettez un fichier `main.cpp` contenant vos tests.

Chapitre VII

Soumission et évaluation par les pairs

Remettez votre devoir dans votre dépôt Git comme d'habitude. Seul le travail effectué dans vos dépôts sera évalué lors de la soutenance. N'hésitez pas à vérifier les noms de vos dossiers et fichiers pour vous assurer qu'ils sont corrects.

Lors de l'évaluation, une brève modification du projet peut être occasionnellement demandée. Il peut s'agir d'un changement de comportement mineur, de quelques lignes de code à écrire ou à réécrire, ou d'une fonctionnalité facile à ajouter.

Bien que cette étape ne soit pas applicable à tous les projets, vous devez vous y préparer si elle est mentionnée dans les directives d'évaluation.

Cette étape vise à vérifier votre compréhension réelle d'une partie spécifique du projet.

La modification peut être effectuée dans n'importe quel environnement de développement de votre choix (par exemple, votre configuration habituelle) et devrait être réalisable en quelques minutes, à moins qu'un délai spécifique ne soit défini dans le cadre de l'évaluation.

On peut par exemple vous demander d'effectuer une petite mise à jour d'une fonction ou d'un script, de modifier un affichage ou d'ajuster une structure de données pour stocker de nouvelles informations, etc.

Les détails (portée, cible, etc.) seront précisés dans les lignes directrices de l'évaluation et peuvent varier d'une évaluation à l'autre pour un même projet.



16D85ACC441674FBA2DF65190663F43A243E8FA5424E49143B520D3DF8AF68036E47
114F20A16827E1B16612137E59ECD492E468BC6CD109F65388DC57A58E8942585C8
D193B96732206