

TP n°1

Introduction à C++

Note : ces exercices nécessitent la lecture des supports des deux premiers cours que l'on peut retrouver à l'adresse <http://www.liafa.univ-paris-diderot.fr/~yunes/cours/cpp/>

Exercice 1 (Hello World)

Comme la tradition le veut, vous devez écrire un programme C++ écrivant un message de bienvenue à l'écran. On rappelle que l'instruction permettant d'envoyer une chaîne de caractères à l'écran a la forme :

```
cout << chaine de caracteres << endl ;
```

sous réserve d'avoir mis en tête de fichier les deux lignes suivantes :

```
#include <iostream>
using namespace std;
```

Utilisez votre éditeur préféré pour le faire. Appelez votre fichier : Hello.cpp.
Compilez avec gcc. Compilez avec g++. Utilisez le Makefile suivant pour la compilation (attention le décalage des lignes de commandes doit correspondre à une tabulation - caractère TAB).

```
GOALS = exo1
```

```
all : $(GOALS)
```

```
exo1 : exo1.o
      g++ -o exo1 exo1.o
```

```
exo1.o : exo1.cpp
      g++ -Wall -c exo1.cpp
```

```
clean :
      rm *.o $(GOALS)
```

Exercice 2 (Code Blocks)

Utilisez l'application Code::blocks (binaire codeblocks) pour faire l'exercice 1. Pour cela, il faut créer un projet en faisant attention de choisir un projet de type « Console Application ». (Tout autre type pouvant potentiellement poser des problèmes par la suite.)

Exercice 3 (Constantes)

Écrivez en C++ une fonction permettant de calculer le sinus d'un angle donné dans une unité parmi (degrés, radians et grades). **Note :** on utilisera des constantes de type `int` pour représenter les unités et, pour pouvoir utiliser la fonction `sin`, on utilisera la directive d'importation de fonction `C` dans le monde C++.

Exercice 4 (Valeurs par défaut)

Ajoutez au code de l'exercice précédent une valeur par défaut permettant de spécifier l'unité par défaut pour exprimer les angles, par exemple, les degrés.

Exercice 5 (Surcharge et initialisation fonctionnelle)

Écrivez un ensemble de fonctions toutes appelées `mon_plus` permettant de calculer : pour l'une la somme de deux ints passés en paramètres et renvoyant un int, pour l'autre la somme de deux doubles et renvoyant un double.

Important : `plus` étant une fonction prédéfinie, ne l'utilisez pas comme nom pour votre fonction.

Appelez la fonction `mon_plus` en lui passant en arguments : deux ints, un int et un short, deux floats, deux doubles, un int et un double. Pour faire ce test, essayez les différents types d'initialisations ("classiques" ou fonctionnelles). Que dit le compilateur ? Pourquoi ?

Exercice 6 (new / delete)

Écrivez une fonction appelée `copie` qui prend en arguments un tableau d'entiers et un nombre d'éléments (un entier `n`) et qui renvoie un nouveau tableau contenant les `n` premiers entiers du tableau.

Exercice 7 (Passage d'un tableau)

Écrivez un ensemble de fonctions, toutes appelées `somme`, et qui permettent de calculer la somme des éléments d'un tableau d'ints pour l'une et de double pour l'autre. Appelez la fonction `somme` en passant : un tableau d'ints, un tableau de shorts, un tableau de doubles. Que dit le compilateur ? Pourquoi ? Comparez avec l'exercice 5.