

C++ 2A

TP 1

1 Hello world

Créer un fichier nommé `main.cpp` contenant le code suivant :

```
#include <iostream>    // inclusion de la gestion des entrées-sorties

int main()             // point d'entrée du programme, retourne un entier pour gérer les erreurs
{
    // Il est possible d'utiliser des opérateurs à la place des fonctions, ici l'opérateur <<
    std::cout << "Hello_C++!" << std::endl;    // Affiche "Hello C++ !"
    return 0;    // retourne 0 car pas d'erreur
}
```

Pour compiler ce code :

```
> g++ -o main main.cpp
```

Pour les codes utilisant le standard C++11 (GCC avant la version 6), il faut ajouter l'option de compilation `-std=c++11`.

Pour l'exécuter :

```
> ./main
```

2 Types

1. Tester les différents types disponibles en C++ : `short`, `int`, `long long`, ... en version signée et non signée, `float` et `double`.
2. Quelle est la taille de chacun de ces types en octets ? Utiliser la macro `sizeof(type)` qui retourne le nombre d'octets utilisés pour représenter le type passé en paramètre pour vérifier.

3 TD1

Coder et tester les exercices du TD1.

4 Procédures et fonctions

1. Créer une fonction factorielle.
2. Créer une procédure prenant en paramètre un tableau, sa taille ainsi qu'une valeur et qui initialise tous les éléments du tableau à cette valeur.
3. Refaire la même chose mais en utilisant un tableau `std::array`, l'avantage est que la taille du tableau est encapsulée dans ce type.
4. Créer une fonction retournant la valeur maximale d'un tableau.
5. Créer une fonction retournant la somme des éléments du tableau.
6. Créer une procédure inversant les éléments d'un tableau.
7. Créer une procédure triant un tableau.

Ajouter le mot clé **const** aux arguments d'une fonction ne devant pas être modifiés dans cette même fonction.

5 class

1. Écrire une classe Personne contenant les noms et prénoms d'une personne.
2. Écrire une méthode toString() à la Java affichant les informations d'une personne.
3. Créer une variable statique privée permettant de compter le nombre de personnes créées, ainsi qu'une méthode statique publique permettant d'accéder à cette information.
4. Ajouter un identifiant à chaque personne en fonction de cette variable statique.
5. Surcharger l'opérateur << afin de pouvoir afficher plus simplement les informations d'une personne.
6. Tester l'opérateur = par défaut entre 2 personnes :

```
Personne p0("Toto", "Titi");
Personne p1 = p0;
cout << p0 << endl;
cout << p1 << endl;
p0.setNom("Tutu");
cout << p0 << endl;
cout << p1 << endl;
```

Quelle est la différence avec Java ?

7. Créer un tableau de 10 personnes.
8. Surcharger l'opérateur < pour pouvoir comparer 2 personnes en fonction de leurs noms.
9. Surcharger l'opérateur << pour afficher une personne.
10. Trier le tableau de personnes en fonction de leurs noms.