

Travaux pratiques N° 1

Les bases du langage C++

Le cours et des documents sont accessibles sur le site :
<https://github.com/Macs1718/CPlusPlus14>.

Exercice 1 (unique_ptr et shared_ptr)

On s'intéresse au bout de code suivant :

```
shared_ptr<int> p_i = make_shared<int>(1);
int a =p_i.use_count();
shared_ptr<int> p_j = share_the_ptr(p_i);
int b =p_i.use_count();
shared_ptr<int> p_k = move_the_ptr(p_i);
int c =p_i.use_count();
int d =p_j.use_count();
```

- Recopier ce code dans un fichier.
- Définir la fonction `share_the_Pointer` qui partagera le pointeur.
- Définir la fonction `move_the_Pointer` qui déplacera le pointeur.
- Exécuter le code. Que valent `a`, `b`, `c`, `d`? On souhaite obtenir 1, 2, 0, 2. Si ce n'est pas le cas, comment modifier votre code?

On s'intéresse maintenant au `unique_ptr` dans le code suivant :

```
unique_ptr<int> p_i = make_unique<int>(1);
unique_ptr<int> p_k = move_the_ptr(p_i);
```

- Recopier ce code.
- Définir la fonction `move_the_Pointer` qui déplacera le pointeur.
- Compiler et exécuter le code.

Exercice 2 (Fonctions et le passage d'arguments)

a) (*Passage d'arguments par valeur*)

Ecrire une fonction `fVal` qui prend un entier `a` comme argument et qui :

- imprime `a` et son adresse,
- y associe la valeur 5,
- l'imprime de nouveau.

Dans la fonction `main`,

- définir une variable `i` de type `int`,
- y associer la valeur 47
- imprimer `i` et son adresse,
- appeler la fonction `fVal` avec `i` comme argument,
- finalement réimprimer `i`.

La valeur de `a` a-t-elle changé dans la fonction `fVal` ? La valeur de `i` a-t-elle changé dans la fonction `main` ?

b) (*Passage d'arguments par référence*) Ajouter une nouvelle fonction `fRef`, où l'argument est passé par référence. L'appeler dans le `main` comme la fonction `fVal`.

c) (*Passage d'arguments par référence constante*) Ajouter une nouvelle fonction `fCRef`, où l'argument est passé par référence constante mais qui cette fois ne fera qu'afficher la valeur. Aurait-on pu appeler cette nouvelle fonction `fRef` ?

d) Dans le code suivant

```
fVal(2);  
fRef(2);  
fCRef(2);
```

Quelles lignes sont correctes ? Expliquer pourquoi.

Exercice 3 (Tableaux statiques et dynamiques)

a) Définir un tableau statique `t` de type `std::array` de 10 entiers.

b) Ecrire une fonction `remplir` qui prend le tableau `t` comme paramètre et qui, à l'aide d'une boucle `for` (style C++ 11) remplit le `t` de sorte que pour tout `i`, `t[i] = 2`.

c) Ecrire une fonction `remplirV2` qui utilisera cette fois une opération prédéfinie pour `std::array`. Consulter votre cours ou <http://en.cppreference.com>.

d) Ecrire une fonction `imprimer` qui prend le tableau `t` comme paramètre et l'imprime. Cette fois-ci, une boucle `for` utilisera les itérateurs.

Exercice 4 (Manipulations de tableaux)

On souhaite écrire la fonction qui, à partir d'une variable `string`, crée un tableau contenant soit

- les termes initialement positifs et 0 à la place des termes négatifs, si la somme des termes du tableau est paire,
- les termes initialement négatifs et 0 à la place des valeurs positifs, si la somme des termes du tableau est impaire.

Cette fonction a la signature suivante :

```
vector<int> compute(const int N, const string& myValues);
```

où `N` correspond au nombre de valeurs contenues dans la chaînes de caractères `Values`.

Prenons deux exemples avec `N = 3` et :

- `myValues = "1 -1 9"`, le résultat attendu est un vecteur contenant 0, -1, 0.
- `myValues = "1 -2 9"`, le résultat attendu est un vecteur contenant 1, 0, 9.

Penser à utiliser la classe `std::istringstream` définie dans la bibliothèque `stream`.

Exercice 5 (Trouver le refuge le plus proche)

Lors d'une randonnée en montagne, la nuit tombe toujours trop tôt. Le but de l'exercice est de donner à partir des coordonnées GPS des randonneurs (longitude et latitude) le nom du refuge le plus proche. L'entrée du programme est fournie dans un fichier formaté de la manière suivante :

- la longitude (`lg`) (en degrés) du randonneur,
- la latitude (`lat`) en degrés) du randonneur,
- Nombre de refuge de la région.
- (*Pour chaque refuge*) `Code Postal; Nom du refuge; localisation; Numéro de téléphone; Longitude (en degrés); Latitude (en degrés)`

Quelques formules pour calculer les distances :

$$distance = 6371 * \arccos [\sin (lat_A) \sin (lat_B) + \cos (lat_A) \cos (lat_B) \cos (lg_B - lg_A)]$$

Remarque :

- comme vous le verrez les coordonnées sont exprimées avec des nombres à virgules. Pensez à transformer les virgules (",") en point(".") pour les applications numériques.
- 6371 correspond au rayon de la terre en km.
- `cos`, `arccos`, `sin` attendent des arguments en radian et sont définies dans la bibliothèque `math.h`.

Écrire un programme `C++` qui :

- a) ouvre et lit le fichier "`Refuges.txt`".
- b) pour chaque refuge, extrait les données pertinentes et calcule la distance entre le randonneur et le refuge.
- c) retourne le refuge le plus proche.

Conseils : vous pouvez entre autre utiliser les classes `string` (`stop`, `substr`, `replace`, `find`, `substr`, ...) , `vector`, ...

Exercice 6 (Bibliographie)

On souhaite écrire une structure `Bibliography` qui permettra de gérer le catalogue d'une bibliothèque. Cette structure contient les champs suivants : `title`, `author`, `year`, `kind`. Le `kind` sera choisi parmi les membres de l'énumération suivante : `Biography`, `Thesis`, `Fiction`, `Fantastic`, `SF`, `Horror`, `Manga`, `Novel`, `Diary`, `Short_story`

- a) Définir une structure `Book` rassemblant les informations dédiées à un livre.
- b) Écrire le corps de la fonction `void add(Bibliographie& bib, const Book& book)` qui ajoute le livre `book` à la bibliographie `bib`.
- c) Définir la fonction `ostream& operator « (ostream& out, const Book& book)` qui affichera les informations associées à un livre passé en argument.
- d) Définir la fonction `void writeBibliography(const string& filename, const Bibliography& biblio)` qui enregistre la bibliographie dans le fichier nommé `filename`.
- e) De même pour la fonction `void readBibliography(const string& filename, Bibliography& biblio)` qui lit la bibliographie enregistrée dans le fichier nommé `filename`.
- f) Finalement écrire les fonctions `const Book& findOldestBook(const Bibliography& bib)` (*resp.* `findNewestBook`) qui trouve le livre le plus ancien (*resp.* récent) de la bibliographie `bib`.