

# MOOC Init. Prog. C++

## Exercices semaine 7

### Exercice 21 : généricité (pointeurs, niveau 1)

Cet exercice correspond à l'exercice n°23 (pages 61 et 228) de l'ouvrage [C++ par la pratique \(3<sup>e</sup> édition, PPUR\)](#).

Un petit exercice simple pour manipuler les pointeurs comme sélecteurs d'objets.

Créez un nouveau fichier appelé `selecteur.cc`.

Dans le `main()`, créez trois variables `valeur1` `valeur2` et `valeur3`, de type `double` que vous initialisez à des valeurs de votre choix.

Déclarez un pointeur `choix`, pointeur sur des `double`.

Demandez ensuite un nombre entre 1 et 3 à l'utilisateur (cf exercice 11 semaine 4) et, en fonction de son choix, faites pointer `choix` sur `valeur1`, `valeur2` ou `valeur3`.

Pour finir affichez `Vous avez choisi` et la valeur choisie **en utilisant le pointeur `choix`**.

Remarque : il est *évident* que le but de cet exercice n'est que de manipuler les pointeurs. Hors de ces contraintes, la bonne solution pour faire la même chose serait bien sûr d'utiliser un tableau.

---

## Exercice 22 : référence (structures, références, niveau 1)

Cet exercice correspond à l'exercice n°26 (pages 64 et 231) de l'ouvrage [C++ par la pratique \(3<sup>e</sup> édition, PPUR\)](#).

Un petit exercice simple pour manipuler les références (C++) comme référencement d'objets (dans le cours, cas numéro 1 d'utilisation des « pointeurs »).

### 22.1 Références

Créez une structure `Maison` contenant simplement une adresse (chaîne de caractères).

Créez ensuite une structure `Personne` contenant un nom (chaîne de caractères) et une *référence* sur une `Maison`.

Les différents habitants d'une maison n'ont en effet pas une *copie* de leur maison avec eux, mais simplement un moyen d'y référer (mémoire de l'adresse)...

C'est un cas typique d'illustration des *références*.

Créez ensuite deux maisons différentes dans le `main()`, puis créez des personnes différentes habitant dans des maisons différentes (plusieurs habitants par maison).

La principale difficulté réside dans le fait qu'une *référence* doit toujours référencer quelque chose : il faut donc affecter les maisons dès l'initialisation.

Pour finir cette partie : créez une fonction `affiche` qui affiche une personne en indiquant son nom et son adresse, par exemple :

```
Pierre habite 12 rue du Château
```

Affichez toutes les personnes de votre programme.

### 22.2 Limites des références (niveau 2)

Le cadre choisi ici illustre aussi les limites des références : avec cette implémentation, il est impossible à une personne de déménager : on ne peut en effet pas « changer de référence ».

C'est parfait si c'est vraiment ce que l'on veut dans le programme (=on est sûr que personne ne déménagera), et il vaudrait alors mieux la déclarer comme référence `constante`.

Mais si l'on voulait pouvoir faire déménager les personnes, il faudrait faire autrement... à vous de trouver comment.

En clair : implémenter une seconde version du programme dans laquelle vous faites déménager au moins une personne.

**Assurez-vous** que les autres personnes à la même (ancienne) adresse **n'ont pas** elles aussi déménagé (ce qui serait le cas avec des références

---