

semaine 6 Tutoriels

MOOC Init Prog C++

Les tutoriels sont des exercices qui reprennent des exemples du cours et dont le corrigé est donné progressivement au fur et à mesure de la donnée de l'exercice lui-même.

Ils sont conseillés comme un premier exercice sur un sujet que l'étudiant ne pense pas encore assez maîtriser pour aborder par lui-même un exercice « classique ».

Semaine 6 : Personnes (structures)

Cet exercice correspond à l'exercice « *pas à pas* » page 45 de l'ouvrage [C++ par la pratique \(3^e édition, PPUR\)](#).

On veut écrire un programme `personne.cc` qui gère (à l'aide de structures) les données relatives à des personnes.

Cet exercice reprend pas à pas les différentes étapes pour y parvenir.

1. Ouvrez le fichier (vide) `personne.cc` dans votre éditeur favori et préparez la « coquille » vide de base accueillant votre programme :

(solution page suivante).

Solution :

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

int main()
{
    return 0;
}
```

2. Déclarez un nouveau type, `Personne`, comme une structure contenant trois champs :
 - `taille` de type réel,
 - `age` de type entier,
 - `sexe` de type caractère.

Remarque : la déclaration d'une structure étant une déclaration de nouveau type (et non pas d'une variable !), elle doit se faire au niveau global du programme, c'est-à-dire avant toute fonction, même le `main()`.

(solution page suivante).

Solution :

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

// déclaration d'une structure
struct Personne{
    double taille;
    int age;
    char sexe;
}; // <-- ATTENTION ! Ne pas oublier le ;

int main()
{
    return 0;
}
```

3. Prototypiez la fonction `affiche_personne` qui aura pour but d'afficher les champs de la structure `Personne`.

(solution page suivante).

Solution :

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

//déclaration d'une structure
struct Personne{
    ...
};

void affiche_personne(Personne const& p); //prototype

int main()
{
    return 0;
}
```

Notez que nous avons ici choisi l'optimisation (plus efficace) consistant à passer une référence (ce qui évite une copie) constante (car la fonction `affiche_personne` ne modifie pas cet argument).

Mais, dans une version tout aussi juste mais peut être un peu moins efficace (ça dépend du compilateur), on aurait aussi bien pu écrire :

```
void affiche_personne(Personne p);
```

4. Définissez maintenant cette fonction (c.-à-d. écrivez son corps).

(solution page suivante).

Solution :

```
#include <iostream>
using namespace std;
...
int main()
{
    return 0;
}

// Affiche les informations d'une personne
void affiche_personne(Personne const& p) {
    cout << "taille : " << p.taille << endl;
    cout << "age : " << p.age << endl;
    cout << "sexe : " << p.sexe << endl;
}
```

5. Prototypiez puis définissez les fonctions anniversaire et naissance. La fonction anniversaire va simplement incrémenter le champ age tandis que la fonction naissance va initialiser tous les champs de la structure (avec les données de l'utilisateur via cin.

(solution page suivante).

Solution :

```
#include <iostream>
using namespace std;
...
void anniversaire(Personne& p) ;
Personne naissance() ;

int main()
{
    return 0;
}
...

// incrémente l'âge d'une personne
void anniversaire(Personne& p) {
    ++(p.age);
}

/* Création d'une personne avec initialisation des champs
 * par l'utilisateur */
Personne naissance() {
    Personne p;

    cout << "Saisie d'une nouvelle personne" << endl;
    cout << "    Entrez sa taille: ";
    cin >> p.taille;
    cout << "    Entrez son age: ";
    cin >> p.age;
    cout << "    Homme [M] ou Femme [F]: ";
    cin >> p.sexe;
    return p;
}
```

6. Compléter le main pour utiliser la structure Personne.

(solution page suivante).

Solution :

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

... // début comme avant

int main()
{

    // Création et initialisation de la structure
    Personne vincent(naissance());

    // Affiche les informations concernant la personne 'vincent'
    affiche_personne(vincent);

    // incrémente l'âge de la personne 'vincent'
    anniversaire(vincent);

    // Ré-affiche les informations concernant la personne 'vincent'
    affiche_personne(vincent);

    return 0;
}
... //suite comme avant
```

Programme complet :

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

// déclaration d'une structure
struct Personne {
    double taille;
    int age;
    char sexe;
};

// prototypes des fonctions
void affiche_personne(Personne const& p);
void anniversaire(Personne& p);
Personne naissance();

// programme principal
int main()
{
    // Création et initialisation de la structure
    Personne vincent(naissance());

    // Affiche les informations concernant la personne 'vincent'
    affiche_personne(vincent);

    // Incrémente l'âge de la personne 'vincent'
    anniversaire(vincent);
}
```

```
// Ré-affiche les informations concernant la personne 'vincent'
affiche_personne(vincent);

return 0;
}

// Définition des fonctions

// Affiche les informations d'une personne
void affiche_personne(Personne const& p){
    cout << "taille : " << p.taille << endl;
    cout << "age : " << p.age << endl;
    cout << "sexe : " << p.sexe << endl;
}

// Incrémente l'âge d'une personne
void anniversaire(Personne& p){
    ++(p.age);
}

/* Création d'une personne avec initialisation des champs
 * par l'utilisateur */
Personne naissance() {
    Personne p;

    cout << "Saisie d'une nouvelle personne" << endl;
    cout << "    Entrez sa taille: ";
    cin >> p.taille;
    cout << "    Entrez son age: ";
    cin >> p.age;
    cout << "    Homme [M] ou Femme [F]: ";
    cin >> p.sexe;
    return p;
}



---


```