

## Exercice 81

---

### Énoncé

Soit la classe `point` suivante :

```
class point
{   int x, y ;
    public :
        point (int abs=0, int ord=0)
        {   x = abs ; y = ord ;
        }
} ;
```

Écrire une fonction indépendante `affiche`, amie de la classe `point`, permettant d'afficher les coordonnées d'un point. On fournira séparément un fichier source contenant la nouvelle déclaration (définition) de `point` et un fichier source contenant la définition de la fonction `affiche`. Écrire un petit programme (`main`) qui crée un point de classe automatique et un point de classe dynamique et qui en affiche les coordonnées.

## Exercice 82

---

### Énoncé

Soit la classe `vecteur3d` définie dans l'exercice 70 par :

```
class vecteur3d
{   float x, y, z ;
    public :
        vecteur3d (float c1=0.0, float c2=0.0, float c3=0.0)
        {   x = c1 ; y = c2 ; z = c3 ; }
        .....
} ;
```

Écrire une fonction indépendante `coincide`, amie de la classe `vecteur3d`, permettant de savoir si deux vecteurs ont les mêmes composantes (cette fonction remplacera la fonction membre `coincide` qu'on demandait d'écrire dans l'exercice 70).

Si `v1` et `v2` désignent deux vecteurs de type `vecteur3d`, comment s'écrit maintenant le test de coïncidence de ces deux vecteurs ?

## Exercice 83

---

### Énoncé

Créer deux classes (dont les membres donnée sont privés) :

- l'une, nommée `vect`, permettant de représenter des vecteurs à 3 composantes de type `double` ; elle comportera un constructeur et une fonction membre d'affichage ;
- l'autre, nommée `matrice`, permettant de représenter des matrices carrées de dimension  $3 \times 3$  ; elle comportera un constructeur avec un argument (adresse d'un tableau de  $3 \times 3$  valeurs) qui initialisera la matrice avec les valeurs correspondantes.

Réaliser une fonction indépendante `prod` permettant de fournir le vecteur correspondant au produit d'une matrice par un vecteur. Écrire un petit programme de test. On fournira séparément les deux déclarations de chacune des classes, leurs deux définitions, la définition de `prod` et le programme de test.