



## TP 3 : Programmation C++

### Exercice 1 :

L'objectif de cet exercice est de gérer les notes des étudiants d'une institution à l'aide d'une classe C++ **Etudiant** définie par :

Les attributs suivants :

- **matricule**: l'identifiant de l'étudiant (**auto incrémenté**)
- **nom**: nom d'un étudiant
- **nbrNotes**: le nombre de notes de l'étudiant
- **\*tabNotes**: tableau contenant les notes d'un étudiant (**allocation dynamique**).

Les méthodes suivantes :

- Un constructeur d'initialisation
- Un constructeur avec arguments
- Un destructeur **~Etudiant ()**
- Un constructeur de copie **Etudiant (const Etudiant &)**
- Les **getters** et **setters**
- **void saisie ()** : permettant la saisie des notes d'un étudiant
- **void affichage ()** : permettant l'affichage des informations d'un étudiant
- **float moyenne ()** : retourne comme résultat la moyenne des notes de l'étudiant.
- **bool admis ()** : retourne comme résultat la valeur **true**, si un étudiant est admis et la valeur **false**, sinon. Un étudiant est considéré comme étant admis lorsque la moyenne de ses notes est supérieure ou égale à 10.

**bool comparer()**: qui compare la moyenne des deux étudiants, retourne comme résultat la valeur **true**, si deux étudiants ont la même moyenne et la valeur **false**, sinon.

### Exemple d'exécution :

```
Creation d'un objet Etudiant E avec 3 notes
```

```
Saisie des notes :
```

```
Donner la note 1 : 11
```

```
Donner la note 2 : 8
```

```
Donner la note 3 : 16
```

```
Affichage de l'etudiant E:
```

```
- Matricule : 1
```

```
- Nom : etud1
```

```
- Nombre de notes : 3
```

```
- Notes : 11 8 16
```

```
Copie de l'etudiant E dans E1
```

```
L'affichage de l'etudiant E1:
```

```
- Matricule : 1
```

```
- Nom : etud1
```

```

- Nombre de notes : 3
- Notes : 11 8 16
Creation d'un objet Etudiant E2 avec 2 notes
Saisie des notes :
Donner la note 1 : 8
Donner la note 2 : 13
Appel des getters :
Le matricule de l'etudiant E2 : 2
Le nom de l'etudiant E2 : etud2
Le nombre de notes de l'etudiant E2 : 2
La moyenne de l'etudiant E est : 11.6667
l'etudiant E est admis
Les Etudiants E et E2 n'ont pas la meme moyenne

```

## Exercice 2 :

Réaliser une classe C++ "vecteur3d" permettant de manipuler des vecteurs à 3 composantes (de type float). On y prévoira :

- un constructeur, avec des valeurs par défaut (0),
- une fonction d'affichage des 3 composantes du vecteur, sous la forme : (x, y, z)
- une fonction permettant d'obtenir la **somme** de 2 vecteurs ;
- une fonction permettant d'obtenir le **produit** scalaire de 2 vecteurs.
- une fonction **coincide** permettant de savoir si 2 vecteurs ont mêmes composantes.
- une fonction qui renvoie la **norme** du vecteur
- une fonction nommée **normax** permettant d'obtenir, parmi deux vecteurs, celui qui a la plus grande norme. On prévoira trois situations :
  - a. le résultat est renvoyé par valeur ;
  - b. le résultat est renvoyé par adresse, l'argument étant également transmis par adresse.
  - c. le résultat est renvoyé par référence, l'argument étant également transmis par référence.

Ecrire un programme utilisant la classe réalisée.

### Exemple d'exécution :

```

Vecteur V1(1,2,3)
Vecteur V2(5,6,7)
La somme des vecteurs v1 et v2 est : (6,8,10)
Le produit scalaire des vecteurs v1 et v2 est : 38
Copier le vecteur V1 dans V3:
Vecteur V3(1,2,3)
Les vecteurs v1 et v3 coïncident
Le vecteur qui a la plus grande norme est (par valeur): (5,6,7)
Le vecteur qui a la plus grande norme est (par adresse): (5,6,7)
Le vecteur qui a la plus grande norme est (par reference) :(5,6,7)

```