



Epreuve : Programmation Orientée Objet C++

Devoir final

2ème année Ingénierie Génie G.M./Ind.

Exercice 1: (les structures)

Définir un type *Date* pour des variables formées d'un numéro de jour, d'un nom de mois et d'un numéro d'année.

1. Ecrire la déclaration d'un type *Fiche* permettant de mémoriser les informations sur un étudiant :
 - ✓ son *nom* ;
 - ✓ son *pre nom* ;
 - ✓ sa date de *Naissance*, de type *Date* ;
 - ✓ sa *formation*, représentée par deux lettres ;
 - ✓ s'il est *redoublant* ou non ;
 - ✓ son *groupe* de TD, représenté par un entier ;
 - ✓ ses notes, représentées par un tableau *note* d'au plus *MAXNOTES* réels ;
 - ✓ un entier *nbnote* indiquant le nombre de notes valides dans le tableau *note*.
2. Ecrire des fonctions de lecture et d'écriture d'une variable de type *Date*.
Dans un premier temps, on ne se préoccupera pas de la validité de la date entrée.
3. Ecrire les fonctions *LireFiche* et *EcrireFiche* de lecture et d'écriture d'une *Fiche*. Aucune note n'est entrée par la fonction *LireFiche*.
4. Ecrire une fonction *AjouteNote* qui reçoit une *Fiche* et ajoute une note, si cela est possible.
5. Ecrire une fonction *Moyenne* qui reçoit une *Fiche* et renvoie, si cela est possible, la moyenne des notes de l'étudiant.

Exercice 2: (les classes)

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 recopie **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
```