

**Série de TP N°7 (DL « Examen de la session  
normale 2020/2021»  
Module : Programmation Orientée Objet en C++**

---

**Consignes :**

- *Aucun document n'est autorisé.*
- *Les téléphones mobiles et autres équipements communicants (exemple : PC, tablettes, etc.) doivent être éteints et rangés dans les sacs pendant toute la durée de l'épreuve.*
- *Sujet de 2 pages.*
- *Le barème est donné à titre purement indicatif. Il est susceptible de changer.*

**Questions de cours (5 points)**

- 1) Expliquer d'une manière générale la différence entre *la liaison statique* et *la liaison dynamique*.
- 2) Etant donnée une classe A en C++, dans quels cas la déclaration « A a=x ; » sera rejetée à la compilation (x étant une variable d'un type quelconque).
- 3) Citer les méthodes qui ne peuvent pas être virtuelles en C++.

**Problème (15 points)**

- N.B.** - *Dans tout le problème utilisez la gestion des exceptions si son utilisation est importante.*  
- *D'autres membres donnés et/ou fonctions membres peuvent être rajoutés aux classes si c'est nécessaire.*

**A) Classe Etudiant**

Considérons une classe `Etudiant` permettant de représenter et gérer des étudiants. Un étudiant est défini par son CNE, nom et prénom, la moyenne générale obtenue pendant l'année en cours et une préférence (3 choix ordonnées concernant une éventuelle option, spécialité ou nom d'une futur école). On doit pouvoir créer un étudiant, l'afficher, lui affecter une préférence ou sa moyenne générale.

- 1) Définir la classe `Etudiant`
- 2) Surdéfinir les opérateurs suivants :
  - `<`, pour comparer deux étudiants et retourner l'adresse de l'objet ayant une moyenne générale maximale
  - `==`, pour tester l'égalité de la moyenne générale de deux étudiants. On doit retourner 1 en cas d'égalité, 0 sinon.

- 3) Etant donné un tableau de  $n$  étudiants, écrire une fonction permettant de retourner l'étudiant ayant une moyenne maximale.

## B) Classe `set_Etudiant`

On s'intéresse à présent à manipuler des ensembles d'étudiants. En utilisant des classes génériques, on doit définir la classe `set_Etudiant` où chaque instance représente l'ensemble des étudiants d'une section donnée.

- 1) Définir la classe `set_Etudiant` avec un constructeur et une fonction d'affichage
- 2) Définir les opérateurs suivants ( $e$  désigne un élément de type `set_Etudiant` et  $n$  un objet de type `Etudiant`)
  - `>>`, tel que `n>>e` ajoute l'élément  $n$  à l'ensemble  $e$  ;
  - `%`, tel que `e%n` vaut 1 si  $n$  appartient à  $e$  et 0 sinon
  - `<<`, tel que `f<<e` envoie le contenu de l'ensemble  $e$  sur le flot indiqué sous la forme `{[objet1], [objet 2],..., [objet n]}`
- 3) Ajouter la fonction `membre cardinal` fournira le nombre d'objets de l'ensemble.

## C) Classe `Ecole`

On veut gérer l'admission des étudiants dans les grandes écoles. L'affectation dans ces écoles est basée sur la moyenne générale et sur la préférence de l'étudiant. Une école est représentée par le nom, la ville, le nombre maximum des étudiants acceptés et la moyenne minimale obtenue par les étudiants admissibles.

- 1) Définir la classe `Ecole` avec un constructeur, une fonction d'affichage et un destructeur.
- 2) Ecrire une fonction `main` permettant la création des objets suivants :
  - `E1("ENSAH", "Al Hoceima", 60, 15.5) ;`
  - `E2("ENSAM", "Meknès", 80, 15.25) ;`
  - `E3("ENSEM", "Casa", 50, 15.75) ;`
- 3) Etant donnée une liste d'étudiants, écrire une fonction permettant de créer un objet `section1` de type `set_Etudiant`.
- 4) Ecrire le programme permettant de faire l'affectation des étudiants de l'objet `section1` aux écoles représentées par les objets `E1`, `E2` et `E3`.