

---

**Devoir Libre n° 1 - Langage C++**  
*A rendre par mail le 09/09/2020 (rapport.pdf)*  
*Il s'agit d'un travail individuel*

**Exercice 1.** Il s'agit de réaliser une application de gestion d'Hôtel. On considère alors que l'hôtel est caractérisé par :

- son nom de type string ;
- son nombre de chambres de type int ;
- le prix d'une chambre de type double.

Le nombre de chambres et le prix d'une chambre sont toujours connus dès la création d'un objet de type Hotel. A l'inverse, il doit être possible de créer un objet Hotel en précisant son nom ou sans le préciser.

Le nombre de chambre d'un hôtel ne sera jamais modifié une fois l'objet construit. A l'inverse, le nom de l'hôtel et le prix d'une chambre doivent pouvoir être modifiés.

Enfin, sur tous les objets Hotel, il faut pouvoir :

- accéder aux valeurs de chacun des attributs ;
- modifier les attributs qui doivent pouvoir être modifiés ;
- récupérer une représentation textuelle de tout l'état de l'hôtel au moyen d'un opérateur :  
    ostream & operator<<(ostream &, const Hotel &) ;
- récupérer le chiffre d'affaire du jour, c'est-à-dire le produit du prix d'une chambre par le nombre de chambres occupées (on suppose que toutes les chambres sont au même prix).

Ecrire la classe Hotel en C++.

Ecrire un petit programme d'essai.

**Exercice 2.** Reprendre l'exercice précédent. On souhaite maintenant modifier la classe Hotel de telle sorte que chaque hôtel connaisse le nombre d'occupants de chacune de ses chambres.

Pour cela, on va ajouter à la classe Hotel un nouvel attribut : un tableau d'entiers qui, dans la ième case, contiendra le nombre d'occupants de la ième chambre.

Réécrivez le code de constructeur de la classe Hotel pour initialiser le nouvel attribut.

Ecrivez :

- le destructeur de la classe Hotel ;
- le constructeur de copie de la classe Hotel ;
- l'opérateur d'affectation de la classe Hotel ;

Ecrivez deux versions de l'opérateur crochet [ ] :

- une version const qui permet d'accéder au nombre d'occupants de la ième chambre de l'hôtel :  
    int operator[](int i) const ;
- une version non const qui permet de modifier le nombre d'occupants de la ième chambre avec une référence :  
    int & operator[](int i) ;

Ecrire un petit programme d'essai.

---

**Devoir Libre n° 2 - Langage C++**

*A rendre par mail le 09/09/2020 (rapport.pdf)*

*Il s'agit d'un travail individuel*

**Exercice 1.** Soit la classe Vect permettant de créer et de manipuler un vecteur :

```
typedef double Real ;
class Vect {
    int n ; // nombre d'éléments du vecteur
    Real *v ; // adresse zone dynamique contenant les éléments
public :
    Vect() ; // constructeur par défaut
    Vect(int) ; // constructeur
    Vect(const Vect&) ; // constructeur de copie
    Vect& operator=(const Vect &) ; // surcharge de l'opérateur =
    ~Vect() ; // destructeur
    Real & operator[](int) const ; // surcharge de l'opérateur [ ] : accès à un élément par son indice
    Vect operator+(const Vect &) const ; // surcharge de + : vecteur + vecteur
    Vect operator*(const Real &) const ; // surcharge de * : vecteur * scalaire
    void init_v(double nbre){ for (int i=0; i < n; i++) v[i] = nbre; } // initialisation d'un vecteur
    friend std::istream & operator>>(std::istream &, Vect&) ; // surcharge de l'opérateur >>
    friend std::ostream & operator<<(std::ostream&, const Vect&) ; // surcharge de l'opérateur <<
private :
    Vect(int, Real*) ; // constructeur
    friend Vect operator*(const Real&, const Vect&) ; // surcharge de * : scalaire * vecteur
} ;
```

Implémenter la classe Vect.

Ecrire un programme d'essai.

**Exercice 2.** Ecrire un programme permettant de réaliser le produit d'une matrice par un vecteur à l'aide d'une fonction indépendante appelée 'produit' amie des deux classes 'matrice' et 'vecteur'.

La classe 'vecteur' possède :

- comme données :

```
int nbelem ; // nombre d'éléments
int *adv ; // adresse zone dynamique contenant les éléments
```

- comme fonctions membres : un constructeur ; un destructeur ; une fonction 'saisie' pour saisir au clavier les composantes d'un vecteur et une fonction 'affiche' pour afficher le contenu du tableau (vecteur).

La classe 'matrice' possède :

- comme données :

```
int nblig ; // nombre de lignes
int nbcol ; // nombre de colonnes
int **adm ; // adresse emplacement dynamique
```

- comme fonctions membres : un constructeur ; un destructeur ; une fonction 'saisie' pour saisir au clavier les composantes d'une matrice et une fonction 'affiche' pour afficher la matrice.