

Exercice 1:

Créer une classe *Personne* représentée par un nom, un prénom et un âge. Le nom et le prénom seront représentés sous forme de tableau dynamique de 20 caractères et l'âge par un entier. On veut pouvoir créer des objets de type *Personne* soit en spécifiant le nom, le prénom et l'âge, soit en ne spécifiant rien. Définir les constructeurs et destructeur associés. Afin de pouvoir tester la validité de votre classe, implémenter une méthode d'affichage : `void affiche() const`.

Exercice 2:

Dans cet exercice vous créerez une classe *Compte* pour gérer le compte bancaire d'un client. On devra retrouver un certain nombre d'informations sur le compte, à savoir :

- _ Le numéro du client qui est unique et attribué automatiquement à la création du compte.
- _ Le nom du client.
- _ le solde.

Ecrire la classe *Compte*, définir les constructeurs et destructeur associés. En donnant les méthodes usuelles d'utilisateur pour la consultation, le retrait et le dépôt.

Exercice 3:

On appellera *dimension* d'une pile le nombre maximal d'éléments qu'elle peut contenir et *taille* d'une pile le nombre d'éléments qu'elle contient réellement. Le tableau représentant la pile est donc indexé de 0 (bas de la pile) à *taille-1* (haut de la pile). L'élément que l'on peut dépiler est donc dans la case d'indice *taille-1*.

Ecrire le fichier *Pile.cpp* correspondant au fichier *Pile.h* suivant :

```
#ifndef Pile H
#define Pile H
class Pile
{ public :
    Pile(int t = 10); // Constructeur qui construit une pile de dimension t (10 par défaut)
    ~Pile(); // Destructeur
    void empile(int); // empile n en haut de la pile
    void depile(); // depile le sommet de la pile
    bool vide() const; // teste si la pile est vide
    bool pleine() const; // teste si la pile est pleine
    int donnetaille() const; // renvoie la taille de la pile
private :
    int dim;
    int taille;
    int *adr;
};
#endif
```