Université Abdelmalek Essaâdi
Faculté des Sciences et Techniques – Tanger
Année Universitaire : 2025/2026
Module : POO en C++, Licence IDAI
Prof. Ikram BEN ABDEL OUAHAB

ATELIER 2

Exercice 1

Écrire une fonction, sans argument ni valeur de retour, qui se contente d'afficher, à chaque appel, le nombre total de fois où elle a été appelée sous la forme : ***appel numéro 3***

Exercice 2

Écrire 2 fonctions à un argument ***entier*** *et une valeur de retour* ***entière*** *permettant de préciser si l'argument reçu est multiple de* ***2*** *(pour la première fonction) ou multiple de* ***3*** *(pour la seconde fonction).*

Utiliser ces deux fonctions dans un petit programme qui lit un nombre entier et qui précise s'il est pair, multiple de 3 et/ou multiple de 6, comme dans cet exemple (il y a deux exécutions) :

```
donnez un entier : 9
il est multiple de 3
---------------
donnez un entier : 12
il est pair
il est multiple de 3
il est divisible par 6
```

Exercice 3

Écrire, de deux façons différentes, un programme qui lit ***10 nombres entiers*** *dans un tableau avant d'en rechercher le plus grand et le plus petit :*

a. en utilisant uniquement le « formalisme tableau » ;
b. en utilisant le « formalisme pointeur », à chaque fois que cela est possible.

Exercice 4

Écrire un programme ***allouant dynamiquement*** *un emplacement pour un* ***tableau d'entiers****, dont la taille est fournie en donnée.*

1. *Utiliser ce tableau pour y placer des nombres entiers lus également en donnée.*
2. *Créer ensuite dynamiquement un nouveau tableau destiné à recevoir les carrés des nombres contenus dans le premier.*
3. *Supprimer le premier tableau, afficher les valeurs du second et supprimer le tout.*

Exercice 5

Ecrire un programme C++ qui :

1. *déclare un entier a;*
2. *déclare une référence vers cet entier ref_a;*
3. *déclare un pointeur vers cet entier p_a;*
4. *affiche les variables, leurs adresses, la valeur pointée.*

Exercice 6

Écrire une fonction nommée ***incrementer****() permettant d'incrémenter la valeur d'une variable passée en paramètre et une fonction nommée* ***permuter****() permettant d'échanger les contenus de 2 variables de type int fournies en argument :*

1. *en transmettant l'adresse des variables concernées (seule méthode utilisable en C) ;*
2. *en utilisant la transmission par référence.*

*Dans les deux cas, écrire un programme (****main****) qui teste les deux fonctions.*

Exercice 7

Écrire une fonction ***récursive*** *en* ***C++*** *qui affiche* ***toutes les permutations possibles*** *d'une chaîne de caractères donnée. Aucune bibliothèque spéciale ne doit être utilisée !!*

Exercice 8

*Vous devez concevoir une classe appelée **Voiture** qui représente les caractéristiques d'une voiture et ses comportements associés. La classe doit inclure les éléments suivants :*

Attributs :

1. *marque (de type std::string) : la marque de la voiture.*
2. *modele (de type std::string) : le modèle de la voiture.*
3. *annee (de type int) : l'année de fabrication de la voiture.*
4. *kilometrage (de type float) : le kilométrage actuel de la voiture.*
5. *vitesse (de type float) : la vitesse actuelle de la voiture.*

Méthodes :

1. *Constructeur par défaut : Initialise tous les attributs avec des valeurs par défaut.*
2. *Constructeur avec paramètres : Permet d'initialiser la voiture avec une marque, un modèle, une année, un kilométrage et une vitesse initiale.*
3. *Méthode accelerer(float valeur) : Incrémente la vitesse actuelle de la voiture en fonction de la valeur passée en paramètre.*
4. *Méthode freiner(float valeur) : Diminue la vitesse actuelle de la voiture en fonction de la valeur passée en paramètre, sans que la vitesse ne devienne négative.*
5. *Méthode afficherInfo() : Affiche les informations sur la voiture (marque, modèle, année, kilométrage, vitesse).*
6. *Méthode avancer(float distance) : Incrémente le kilométrage en fonction de la distance passée en paramètre.*
7. *Destructeur : Affiche un message indiquant que la voiture est détruite.*

Exercice 9

*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 :

- *le résultat est renvoyé par valeur ;*
- *le résultat est renvoyé par adresse, l'argument étant également transmis par adresse.*
- *le résultat est renvoyé par référence, l'argument étant également transmis par référence.*

Exercice 10

*Effectuer les opérations arithmétiques sur des **nombres complexes** à l'aide d'une classe et d'un objet.*
Le programme doit demander la partie réelle et imaginaire de deux nombres complexes et afficher les parties réelle et imaginaire de l'opération demandée. (égalité, addition, soustraction, multiplication, division). Le choix de l'opération peut être fait par un Menu.