



TP6 - Programmation Orientée Objet en C++ avec Pointeurs

Imad Kissami

3 Mars 2025

Instructions

- Créer un `Makefile` pour compiler tous les exercices.
- Chaque exercice doit être sauvegardé sous le format : `exo{numero_exercice}.cpp`.
- L'exécution se fait avec : `./exo1` (pour l'exercice 1).
- Tous les fichiers doivent être regroupés dans un dossier `TP6_Nom_Prénom`.

Exercice 1 : Définition d'une classe et création d'objets

Objectif :

- Définir une classe `Joueur` avec des attributs (`nom`, `score`).
- Implémenter des méthodes pour afficher les informations du joueur.
- Créer plusieurs objets et les manipuler.

Output attendu :

Joueur: Alice, Score: 100
Joueur: Bob, Score: 200

Exercice 2 : Gestion d'objets avec pointeurs

Objectif :

- Démontrer l'utilisation des modificateurs d'accès (`public`, `private`).
- Utiliser les opérateurs `.` et `->` pour accéder aux membres d'une classe.
- Allouer dynamiquement un objet avec `new` et le libérer avec `delete`.

Output attendu :

Nom: Alice
Score: 100
Objet dynamique supprimé.

Exercice 3 : Constructeurs et Destructeurs

Objectif :

- Implémenter un constructeur qui initialise les attributs d'un objet.
- Ajouter un destructeur affichant un message lors de la destruction d'un objet.

Output attendu :

Constructeur appelé
Joueur: Alice, Score: 100
Destructeur appelé

Exercice 4 : Gestion d'un inventaire avec pointeurs dynamiques

Objectif :

- Définir une classe `Objet` avec un `nom` et une `quantité`.

- Créer une classe **Inventaire** stockant des objets sous forme de **pointeurs dynamiques**.
- Implémenter des méthodes pour ajouter et retirer des objets.
- Libérer la mémoire allouée.

Output attendu :

```
Ajout de Pomme (x3)
Ajout de Épée (x1)
Inventaire :
- Pomme (x3)
- Épée (x1)
Mémoire de l'inventaire libérée.
```

Exercice 5 : Gestion d'un système de notes

Objectif :

- Définir une classe **Étudiant** avec un nom et une liste de notes.
- Implémenter une méthode pour calculer la moyenne des notes.

Output attendu :

```
Nom: Karim
Notes: 15, 18, 12
Moyenne: 15.0
```

Exercice 6 : Manipulation de fichiers avec pointeurs

Objectif :

- Définir une classe **Fichier** permettant de lire et écrire dans un fichier texte.
- Utiliser un **pointeur vers un flux de fichier** pour manipuler les données.
- Implémenter des méthodes pour sauvegarder et charger des données.

Output attendu :

```
Sauvegarde des données...
Lecture des données...
Contenu du fichier :
Bonjour, ceci est un test.
Mémoire du flux libérée.
```

Exercice 7 : Implémentation de `shared_ptr` (sans utilisation du smart pointer `Shared_ptr`)

Objectif :

- Implémenter une classe **SharedPtr** qui gère un pointeur avec un compteur de références.
- Gérer l'incrément et la décrémentation du compteur lors de la copie et de la destruction.
- Assurer la suppression automatique de l'objet lorsque le compteur atteint zéro.

Output attendu :

```
Création d'un SharedPtr vers 10
Nombre de références : 1
Création d'une copie
Nombre de références : 2
Destruction d'un SharedPtr
Nombre de références : 1
Destruction d'un SharedPtr
Objet supprimé
```