

# Mini-Projet C++ : Gestion d'une Bibliothèque (POO)

## Contexte

Vous etes en charge de developper une application console permettant de gerer une petite bibliotheque.

Cette application devra permettre d'ajouter des livres et des utilisateurs, d'effectuer des emprunts et des retours de livres, et d'afficher les informations disponibles.

## Concepts a maitriser

- Classe
- Encapsulation (attributs prives, getters, setters)
- Constructeurs et destructeurs
- Heritage (classes de base et derivees)
- Polymorphisme (methodes virtuelles)
- Composition (une bibliotheque contient des livres)
- Gestion dynamique de memoire (new/delete)
- Surcharge d'operateurs
- Vecteurs dynamiques (std::vector)
- Surcharge de methodes
- Gestion d'exceptions (bonus)

## Details des classes

### 1. Classe Livre

- Attributs prives : titre, auteur, isbn, disponible (bool).
- Methodes :
  - afficherInfos()
  - emprunter()
  - rendre()
  - Getters et setters.

### 2. Classe Utilisateur

## Mini-Projet C++ : Gestion d'une Bibliothèque (POO)

- Attributs privés : nom, prenom, id.

- Methodes :

- afficherInfos() (methode virtuelle)

- emprunterLivre(Livre&)

- rendreLivre(Livre&)

### 3. Classes derivees Etudiant et Professeur

- Heritent de Utilisateur.

- Different nombre maximum de livres empruntables :

- Etudiant : 3

- Professeur : 5

- Redefinir afficherInfos() pour preciser le type d'utilisateur.

### 4. Classe Bibliotheque

- Attributs :

- std::vector<Livre\*> livres

- std::vector<Utilisateur\*> utilisateurs

- Methodes :

- ajouterLivre()

- ajouterUtilisateur()

- afficherLivres()

- afficherUtilisateurs()

- chercherLivreParISBN()

- emprunterLivre(utilisateurID, isbn)

- rendreLivre(utilisateurID, isbn)

### Fonctions supplementaires (Bonus)

- Exception LivreNonDisponibleException si tentative d'emprunt d'un livre deja pris.

- Surcharge de l'operateur == pour comparer deux livres via l'ISBN.

# Mini-Projet C++ : Gestion d'une Bibliothèque (POO)

## Menu Console Exemple

---- Menu Bibliotheque ----

1. Ajouter un livre
2. Ajouter un utilisateur
3. Emprunter un livre
4. Rendre un livre
5. Afficher tous les livres
6. Afficher tous les utilisateurs
0. Quitter

## Contraintes Techniques

- Le code doit etre proprement commente.
- Utiliser std::vector pour la gestion des collections.
- Respecter les bonnes pratiques de la POO.
- Gestion memoire propre (delete sur tous les new).

## Livraison attendue

- Tous les fichiers .cpp et .h necessaires.
- Un fichier main.cpp pour le menu d'interaction.

Bonne programmation !

"La ou d'autres se contentent d'apprendre, toi, bâtis un chef-d'oeuvre."