

## TP P.O.O

### Exercice I:

#### Objectif:

Le but de ce TP est de pratiquer la création de classes en C++ et d'implémenter un constructeur par copie en utilisant la copie profonde, avec l'utilisation de **new** et **delete** pour la gestion dynamique de la mémoire

#### Travail demande:

La classe Adresse a les attributs suivants :

- Un attribut **rue**
- Un attribut **numéro**

Implémentez la classe Adresse avec les caractéristiques suivantes :

Deux attributs : char\* pour la rue et int pour le numéro.

1. Un constructeur par défaut qui initialise la rue à nullptr et le numéro à 0.
2. Un constructeur paramétré qui prend une rue et un numéro en arguments et initialise les attributs correspondants avec des copies dynamiques.
3. Un constructeur par copie qui crée une copie profonde de la rue de l'objet source.
4. Un destructeur qui libère la mémoire allouée pour la rue.
5. Dans la fonction main, créez une instance de chaque classe, utilisez les constructeurs par copie pour créer des copies de ces instances, puis affichez les informations originales et copiées.

### Exercice II:

#### Objectif:

Vous devez créer une classe C++ appelée "Personne" pour représenter une personne avec un nom et un âge. Ensuite, vous devez créer une classe "ListePersonnes" qui implémente une liste chaînée de personnes. La difficulté réside dans la création d'un constructeur de copie qui effectue une copie profonde de la liste, c'est-à-dire qu'elle crée de nouvelles instances de "Personne" pour chaque personne de la liste.

**Travail demande:**

1. Implémentez la classe "Personne" avec les attributs nom et âge, ainsi qu'un constructeur par défaut et un constructeur paramétré pour initialiser ces attributs.
2. Implémentez la classe "ListePersonnes" qui gère une liste chaînée de personnes. Cette classe doit avoir des méthodes pour ajouter une personne à la liste, afficher la liste, et une méthode de destructeur pour libérer la mémoire utilisée.
3. Créer un constructeur de copie pour la classe "ListePersonnes" qui effectue une copie profonde de la liste. Assurez-vous de ne pas partager les mêmes instances de personnes entre les deux listes.
4. Dans la fonction main(), créez une liste de personnes et ajoutez-y quelques personnes. Ensuite, utilisez le constructeur de copie pour créer une nouvelle liste en copiant la première. Assurez-vous que les modifications apportées à la nouvelle liste n'affectent pas la liste d'origine.
5. Testez votre programme en ajoutant, supprimant ou modifiant des personnes dans l'une des listes et vérifiez que l'autre liste reste inchangée.