

## **TD25 - Pointeurs intelligent**

### **Exercice clarté : Lisibilité, commentaire, accessibilité (1pts)**

Lors de votre correction, observer le code, la façon dont il est construit, les commentaires, la lisibilité. En fonction de tous ces critères, retirer des points ou non sur les 2.

### **Exercice 1 : Introduction aux unique\_ptr (4pts)**

Produire un programme qui définit un unique pointer. Il doit stocker une valeur int. Dans un deuxième temps, templater le pointeur et afficher le contenu. Pour finir, définir un deuxième unique pointer et récupérer le contenu du précédent unique\_ptr. Afficher les deux uniques. Que constatez-vous?

### **Exercice 2 : Introduction au bloc de contrôle (3pts)**

Produire un programme qui met en place 3 pointeurs partagés qui pointent vers le même objet. Effectuez des tests en supprimant, recréant et modifiant des informations. Que constatez-vous?

### **Exercice 3 : pointeurs intelligent avancés (3pts)**

Produisez la structure de votre choix qui doit contenir plusieurs shared pointer. Elle doit être templaté et permettre le mouvement.

### **Exercice 4 : RPG return! (3pts)**

Intégrer un personnage qui possède plusieurs systèmes liés à d'autres classes via des agrégations. (au moins 3). Utiliser ce qui a été vu en cours.

### **Exercice 5 : Ownership (3pts)**

Produire un programme qui définit un pointeur nu et un pointeur faible. Donnez de la data à ces deux pointeurs. Puis rattacher le weak pointer à des shared pointer. Delete l'objet ou la data pointé. Que constatez-vous? Quelle est donc la différence fondamentale entre le shared\_ptr et le weak\_ptr? En quoi un weak\_ptr devient-il plus pertinent qu'un raw pointer dans ce cas précis?

### **Exercice 6 : Pour aller plus loin (3pts)**

Nous avons abordé le concept d'idiome lors du cours sur les pointeurs intelligents. Citez au moins deux autres idiomes et présentez les lors d'une courte description (10 lignes max)