

## ***L2 DE C++***

**Mercredi 13 mai 2015 – Durée 2h**

### **Documents non autorisés**

#### **Consignes à suivre impérativement :**

- 1) Eviter les ratures sur les copies et soigner votre présentation et votre écriture. La copie doit être lisible afin de pouvoir être corrigée.
- 2) Utiliser à bon escient des références et le mot-clé const

#### **Exercice 1 : Pointeurs et références**

**2,5 points**

Soit la fonction *exercice\_pointers* dont le code est donné ci-dessous :

```
void exercice_pointers () {  
    int * p;  
    int a = 7;  
    int b = 27;  
    int x = 2;  
    int y = 3;  
  
    p = &a;  
    cout << "a = " << *p << endl;  
    *p = *p + y;  
    cout << "a = " << *p << endl;  
    p = &x;  
    cout << "x = " << *p << endl;  
    y = y + 5 * (*p);  
    cout << "y = " << y << endl;  
}
```

- 1) Donner le résultat de l'exécution de cette fonction.
- 2) Réécrire le code ci-dessus en modifiant le **pointeur p** par une **référence ref** et en renommant la fonction ***void exercice\_references(void***

## Exercice 2: classe Musicien

**4 points**

Soit la **classe Musicien**, définie dans un fichier **musicien.h**, dont les attributs sont donnés ci-après :

```
class Musicien {  
private :  
    string a_nom;  
    Instrument a_instrument;  
    Date a_naissance;  
public :  
    ....  
};
```

Les **classes Date et Instrument** sont supposées être existantes et contenir les fonctions et opérateurs nécessaires, notamment les constructeurs par défaut et de copie.

Dans un fichier **musicien.cpp** :

- 1) Ecrire la définition du constructeur à 3 paramètres. L'utilisation d'une liste d'initialisations est requise.
- 2) Ecrire la définition du constructeur de copie.
- 3) Deux musiciens souhaitent échanger leur instrument.  
Ecrire la définition de cette fonction d'échange :
  - a) en tant que fonction membre de classe
  - b) en tant que fonction externe à la classe

### **Exercice 3 : classe Arbre**

**8 points**

1) Définir une **classe Arbre** dans un fichier **arbre.h** où figure toutes les définitions des fonctions membres évoquées ci-après.

Un arbre a une taille et un âge dont les valeurs par défaut sont respectivement 175 cm et 20 ans.

- a) Est-il nécessaire de définir un constructeur de copie et pourquoi? Dans l'affirmative, écrire sa définition.
- b) Est-il nécessaire de définir une surcharge de l'opérateur d'affectation et pourquoi? Dans l'affirmative, écrire sa définition.
- c) Est-il nécessaire de définir le destructeur et pourquoi? Dans l'affirmative, écrire sa définition.
- d) Les fonctions membres doivent permettre :
  - d'initialiser un arbre avec les valeurs par défaut
  - de consulter ses 2 attributs et de les mettre à jour (les accesseurs en lecture et en écriture)
  - de l'afficher via une surcharge d'opérateur sous la forme (180 21) pour un arbre de taille 180 cm et d'âge 21 ans.

2) Ecrire la fonction principale **main** dans un fichier **ArbreTest.cpp**, réalisant les services suivants grâce aux fonctions définies ci-dessus :

- création d'un premier arbre de taille 175 cm et d'âge 20 ans (allocation statique)
- création d'un second arbre de taille 180 cm et d'âge 21 ans (allocation dynamique)
- changement de l'âge du second arbre : 23 ans
- affichage des 2 arbres.

**Attention :** ne pas oublier d'inclure les fichiers d'entête nécessaires.

#### **Exercice 4 : classe Musicien**

**4 points**

Soit la **classe Musicien**, définie dans un fichier **musicien.h**, dont les attributs sont donnés ci-après :

```
class Musicien {  
private :  
    char * a_nom;  
    int a_naissance;  
    int a_mort;  
    char * a_instrument;  
public :  
    ....  
};
```

Les définitions des fonctions demandées doivent figurer dans un fichier **musicien.cpp**.

- 1) Ecrire la définition du constructeur à 4 paramètres.
- 2) Est-il nécessaire de définir un constructeur de copie et pourquoi? Dans l'affirmative, écrire sa définition.
- 3) Est-il nécessaire de définir une surcharge de l'opérateur d'affectation et pourquoi? Dans l'affirmative, écrire sa définition.
- 4) Est-il nécessaire de définir le destructeur et pourquoi? Dans l'affirmative, écrire sa définition.

#### **Exercice 5: Classe abstraite**

**0,5 point**

Qu'est-ce qu'une classe abstraite ? Comment la met-on en oeuvre en C++ ?

#### **Exercice 6: Polymorphisme**

**1 point**

Qu'est-ce que le polymorphisme ? Comment le met-on en oeuvre en C++ ?