

Epreuve de Moyenne Durée

Exercice N° 1: (4 PTS)

a. Soit la déclaration suivante:

```
1  class Rect
2  {
3      int x,y;
4      unsigned int width, height;
5  public:
6      virtual void setWidth(unsigned int w) { width=w; }
7      ...
8  };
9
10 class Square: public Rect
11 {
12 public:
13     void setWidth(unsigned int w) { width = height = w; }
14     ...
15 };
16
17 int main()
18 {
19     Rect* obj = new Square();
20     obj->setWidth(100);
21 }
```

Dans la ligne 19: **obj** est il un **Square** ou un **Rect** ? (1 pt)

Dans la ligne 20: quelle méthode **setWidth** est appelée (**Rect::setWidth** ou bien **Square::setWidth**) dans les deux cas suivants:

- le mot **virtual** précède la définition de cette méthode. (1 pt)
- il n'y a pas le mot **virtual**. (1 pt)

b. Donner la sortie du programme suivant: (1 pt)

```
.....
class exemple {
static int x; int y;
public :
    exemple(int y) { exemple::y=y; x++; };
    void affiche() { cout<< "Valeur de y = " << y << " ; valeur de
x = " << x << endl; };
};

int exemple::x=0;

int main() {

    exemple e1(3), e2(6);
    e1.affiche();
    e2.affiche();

    return 0;
}
```

Exercice N° 2: (7 pts)

Soient les quatre classes suivantes:

- Une classe **X** dont les membres sont: un entier **A** et une méthode **set_A** qui met **A** à 0

- Une classe **Y** qui hérite de **X** et dont les membres sont: une méthode **f()** qui instancie 2 objets: un objet **x** de la classe **X** et un objet **y** de la classe **Y** elle même. Cette méthode met à **1** le membre **A** de **x** et à **1** le membre **A** de **y**.
 - Une classe **Z** qui hérite de **Y** et dont les membres sont: une méthode **g()** qui instancie 3 objets: un objet **x** de la classe **X**, un objet **y** de la classe **Y** et un objet **z** de la classe **Z** elle même. Cette méthode met à **2** le membre **A** de **x**, à **2** le membre **A** de **y** et à **2** le membre **A** de **z**.
 - Une classe **U** dont les membres sont: une méthode **h()** qui instancie 3 objets: un objet **x** de la classe **X**, un objet **y** de la classe **Y** et un objet **z** de la classe **Z**. Cette méthode met à **3** le membre **A** de **x**, à **3** le membre **A** de **y** et à **3** le membre **A** de **z**.
1. Donner le schéma d'héritage de ces classes. (1 pt)
 2. Notez dans un tableau les droits d'accès pour les cas suivants: (3 pts)
 - Cas 1: **A private** dans **X**, et **Y** et **Z** dérivent comme suit: **class Y: X** et **class Z: Y**
 - Cas 2: **A private** dans **X**, et **Y** et **Z** dérivent comme suit: **class Y: public X** et **class Z: Y**
 - Cas 3: **A public** dans **X**, et **Y** et **Z** dérivent comme suit: **class Y: X** et **class Z: Y**
 - Cas 4: **A public** dans **X**, et **Y** et **Z** dérivent comme suit: **class Y: public X** et **class Z: Y**
 - Cas 5: **A protected** dans **X**, et **Y** et **Z** dérivent comme suit: **class Y: X** et **class Z: Y**
 - Cas 6: **A protected** dans **X**, et **Y** et **Z** dérivent comme suit: **class Y: public X** et **class Z: Y**
 - Cas 7: **A public** dans **X**, et **Y** et **Z** dérivent comme suit: **class Y: public X** et **class Z: public Y**

voici un exemple de tableau: (**O** signifie Oui accessible, **N** signifie Non accessible)

Accès	Dans Classe		
	Y	Z	U
x.A	O	O	N
y.A	O	O	N
z.A		O	N

3. Reprendre la question précédente pour le cas où **U** est amie de **X**. (3 pts)

Exercice N° 3: (9 pts)

Le but est d'écrire un programme de gestion de paie pour une entreprise employant plusieurs travailleurs de différentes catégories:

Nous considérons les classes suivantes:

- Une classe de base **EMPLOYE** dont les attributs sont le **NOM** et **PRENOM**, un constructeur et un destructeur.
 - Une classe **CHEF** qui dérive d'**EMPLOYE** et qui dispose d'un salaire fixe et d'une prime de responsabilité qui est un pourcentage sur les ventes.
 - Une classe **SOUS_CHEF** qui dérive de **CHEF** et qui fait des heures supplémentaires pour lesquelles il est payé.
 - Une classe **EMPLOYE_COMMISSION** qui reçoit une prime fixe et un pourcentage sur ses ventes.
- Toutes les classes disposent d'une fonction **GAIN**. Elle se charge de calculer les gains de chaque employé.
 - Une fonction d'affichage des informations de l'employeur est commune à toutes les classes.
 1. À partir des informations données plus haut définir ces classes en complétant les attributs nécessaires à chaque classe. (5 pts)
 2. Justifier l'utilisation du caractère virtuelle pure de la fonction **GAIN** et virtuelle de la fonction d'affichage. (1 pt)
 3. Conclure sur la classe **EMPLOYE**. (1 pt)
 4. Le programme principal instancie des exemples d'employés et affiche leurs informations (nom, prénom et gains). (2 pts)