Correction examen (2018/2019) : L2-S3	Module : POO	Durée : 01h30	GACEB

Exercice 1 (6 points):
Ecrire la définition de la classe Cube qui encapsule :

1. **(0.5 pt)** un tableau T (des entiers) de taille 3D L×C×P (L, C et P sont aussi des attributs de cette classe),

```
class Cube{  public:
   int ***T, L, C, P;
   //Les méthodes des questions 2 à 6 seront insérées ici
};
```

2. **(1 pt)** un constructeur permettant d'allouer dynamiquement la mémoire nécessaire pour T selon la taille renseignée par ses trois arguments d'entrée (l, c et p) avec une valeur par défaut =1 ,

```
Cube(const int l=1, const int c=1, const int p=1){
        L=l; C=c; P=p;
        T=new int **[L];
        for(int i=0; i<L; i++) {
            T[i]=new int *[C];
            for(int j=0; j<C; j++)
             T[i][j]=new int [P] ;
        }
}
```

3. **(0.5 pt)** un destructeur permettant une libération complète de la mémoire allouer pour T,

```
~Cube() {
       for(int i=0; i<L; i++) {
            for(int j=0; j<C; j++)
              delete [] T[i][j];
            delete [] T[i];
        }
        delete [] T;
}
```

4. **(1 pt)** un opérateur de copie

```
Cube & operator= (const Cube & cb) {
   // Désallouer l'espace existant
        for(int i=0; i<L; i++) { for(int j=0; j<C; j++) delete [] T[i][j];
            delete [] T[i];}
        delete [] T;
    // Allouer un nouveau espace
      L = cb.L; C = cb.C; P=cb.P;
      T=new int **[L];
      for(int i=0; i<L; i++){ T[i]=new int *[C];
         for(int j=0; j<C; j++){ T[i][j]=new int [P] ;
            for(int k=0; k<P; k++) T[i][j][k]=cb.T[i][j][k];
         }}
 return *this;
}
```

5. **(1 pt)** un constructeur de copie

```
Cube(const Cube & cb){
      L = cb.L; C = cb.C; P=cb.P;
      T=new int **[L];
      for(int i=0; i<L; i++){ T[i]=new int *[C];
         for(int j=0; j<C; j++){ T[i][j]=new int [P] ;
            for(int k=0; k<P; k++)  T[i][j][k]=cb.T[i][j][k];
         }}
}
```

6. **(2 pt)** une méthode Somdiag qui affiche la somme des éléments de la diagonal de T si L=C=P, sinon, elle affiche la valeur « -1 ». Il est interdit d'utiliser des boucles (for, while et do while), pas de goto aussi.

```
   int SM(int n){  if (n < 0)  return 0; else return T[n][n][n] + SM(n-1); }
   void Somdiag() {   if ((L==C)&&(L==P)) cout<< SM(L-1);  else cout<<"-1"; }
```

Exercice 2 (4 points):

1. Imaginer un exemple de votre choix (pratique) d'utilisation d'une classe qui encapsule trois attributs, chacun est de type générique différent (il est interdit de donner le même exemple vu en cours, TD ou TP). Donner la définition et la déclaration de cette classe ainsi qu'un exemple de son utilisation dans la fonction main().

(0.5 pt)

```
template <class T, class U, class V>
 class A {
 public:
    T a;  U b;  V c;
     A(T a=0, U b=0, V c=0): a(a), b(b), c(c) { }
 };
```

(0.5 pt)

```
int main()
{
    A<int, double, string> ob(7,7.6,"Informatique");
    Cout<< ob.a<<","<<ob.b<<","<<ob.c ;
}
```

Attention. Compter comme juste toute autre exemple donné jamais vu en cours, td ou TP
Si l'étudiant donne un exemple correcte déjà vu en td ou tp sa note est 0

2. Ajouter à cette classe une variable statique, proposer un cadre applicatif de ce type de variable (adapté à votre exemple de la question 1), donner le code permettant d'initialiser et de faire évoluer cette variable.

(1 pt)

```
template <class T, class U, class V>
 class A  {
 public:
    T a;  U b;  V c;
    static int cpt; // utilisée comme compteur d'objet

    A(T a=0, U b=0, V c=0): a(a), b(b), c(c) { cpt++; }
    ~A(){ cpt--;}
 };
```

(0.5 pt)

```
template <class T, class U, class V>
 int A<T, U, V>::cpt=0;
```

(0.5 pt)

```
int main(){

    A<int, double, string> a,b,c;
    cout<<"nombre objets ="<<a.cpt;
}
```

Exercice 3 (6 points)

Si les classes Banane et Raisin dérivent de la classe Fruit et la classe Cardinal dérive de la classe Raisin.

1. Ecrire la déclaration de ces quatre classes

```
class fruit {};
class banane: public  fruit{} ;
class raisin: public  fruit{} ;
class cardinal: public  raisin {};
```

(1pt)=0.25pt *pour toute classe juste (toute erreur dans l'écriture d'une classe conduit à la perte de 0.25pt*

2. Toute classe doit contenir la méthode «Afficher » qui affiche le nom de fruit concerné.

```
class fruit {public: virtual void afficher(){cout<<"\n Fruit";}};
class banane: public  fruit { public: void afficher(){cout<<"\n Banane";} };
class raisin: public  fruit { public: void afficher(){cout<<"\n Raisin";} };
class cardinal: public raisin{public: void afficher(){cout<<"\n Cardinal";}};
```

(1pt)=0.25pt *pour toute classe juste (toute erreur dans l'écriture d'une classe conduit à la perte de 0.25pt*

3. Indiquer si chacune des instructions suivantes est vraie ou fausse (expliquer pourquoi) :
   a. Fruit *tab = new Banane [3] ; ***Correcte*** *création d'un tableau fruit contenant trois objets banane*
   b. Fruit *tab = new Fruit [3] ; ***Correcte*** *création d'un tableau fruit contenant trois objets fruit*
   c. Fruit **tab = new Fruit *[3]; ***Correcte*** *création d'un tableau de trois pointeur de type fruit*
   d. Cardinal *tab = new Raisin [3] ; ***Faut*** *Conversion raisin vers cardinal est invalide*
   e. Cardinal * tab = new Banane [3] ; ***Faut*** *Conversion banane vers cardinal est invalide*

   (1.25 pt) = 0.25pt *pour toute réponse juste, Absence de justification = perte de la moitié de note*

4. Parmi les cinq propositions ci-dessus, lesquelles permettent de créer un tableau tab pouvant contenir les trois objets suivants (dans cet ordre et en même temps) : Banane, Raisin et Cardinal ?
   *Rép : la proposition c (Fruit **tab = new Fruit *[3])*
   (0.75 pt) *: Toute autre proposition conduit à 0*

5. Ecrire un code permettant de charger ces trois objets dans un même tableau tab (selon chacune des propositions retenues dans la question 1) et afficher le nom de chaque fruit en parcourant tous les éléments de tab.

```
Fruit **tab = new Fruit *[3];
tab[0] = new banane();    (0.25pt)
tab[1] = new raisin();    (0.25pt)
tab[2] = new cardinal();  (0.25pt)

for(int i=0; i<3; i++)
  tab[i]->afficher();      (0.25pt)
```

(1pt)

6. Donner une solution permettant de créer Tab, de le charger et d'afficher son contenu en utilisant la bibliothèque STL.

```
#include <vector>                (0.25pt)
…
vector <fruit *> tab;            (0.25pt)
tab.push_back(new banane());
tab.push_back(new raisin());      (0.25pt)
tab.push_back(new cardinal());

for(int i=0; i<tab.size(); i++)   (0.25pt)
   tab[i]->afficher();
```

(1pt)

Exercice 4 (4 points)

Le programme suivant comporte des erreurs. Indiquer le numéro de toute ligne comportant une erreur avec une brève explication et un code correcte.

Les ajout sont marquée en couleur

```
1.  class A {
2.    public :                  // ajout deux points ici
3.    int a;
4.    virtual void f1()=0;     // ajout mot virtual avant cette méthode abstrait pure
5.    virtual {} void f2()=0;      // ajout mot virtual et suppression {}
6.    };                        // ajout le ;
7.  class B : public A { // ajout :
8.    public :
9.  int b;     void f1(){} ;    void f2(){} ; // ajout obligatoire la défintion de f1 et f2
    car dans A étaient virtuelles pures
10. };     // ajout le ;
11. int main() {       // ajout accolade  ici
12. A * a1;   // erreur A est abstraite ne peut pas être instanciable, mais un pointeur peut
    être créer de A
13. A a2=new A();   //  Erreur A est abstraite ne peut pas être instanciable,
14.
15. A *a3=new B(); // erreur  a3 doit être un pointeur pour assurer une liaison dynamique
16. B b1;
17. B b2= new A(); // conversion impossible, A est abstraite n'est pas inctanciable
18. return 0;
19. }
```

0.25 pt pour chaque erreur signalée, corrigée et expliquée

Bon courage