

NOM: Prénom: Groupe TD:

Examen d'informatique 2^{ème} Année
Ecole Centrale de Lyon - département MI
(Lundi 15 décembre 2003 - durée 2h00)

Les réponses seront faites directement dans les emplacements réservés à cet effet sur le sujet qui sera rendu à la fin de l'examen. **Aucun document (support de cours, TD, etc...) n'est admis (sauf dictionnaires de langues pour les étrangers). Les calculatrices sont également interdites.**

QCM (Questions à Choix Multiples) : Répondre en cochant la ou les cases correspondantes aux affirmations qui sont exactes et **ne cochant pas** les affirmations qui sont fausses.
[Notation par question : réponse juste : +1/2 point ; réponse fausse : -1/2 point ; pas de réponse : 0 point]

Q1: En C++, **this** désigne :
(1/2 pt) le nom de la classe
 un pointeur sur l'instance de la classe
 une référence sur la classe
 le nom de la fonction qui est amie de la classe

Q2: Une fonction amie (**friend**) d'une classe a accès :
(1/2 pt) aux attributs privés de cette classe
 aux méthodes privées de cette classe
 aux attributs protégés de cette classe
 aux méthodes protégées de cette classe
 aux attributs publics de cette classe
 aux méthodes publiques de cette class

Q3: Pour une classe, si l'utilisateur ne les définit pas,
(1/2 pt) le compilateur génère une version par défaut (ou autorise l'utilisation) pour :
 le constructeur sans paramètre
 le constructeur à 1 paramètre entier
 le constructeur par copie
 le destructeur
 l'opérateur =
 l'opérateur ++

Q4: Les opérateurs ci-dessous peuvent être surchargés
(1/2 pt) en C++ :
 operator= : opérateur binaire d'affectation
 operator[] : opérateur binaire d'accès à une valeur d'un tableau par un indice
 operator new : opérateur unaire d'allocation dynamique
 operator?: : opérateur ternaire de test
 aucun des opérateurs ci-dessus

Q5: En C++, par héritage, sont hérités dans la classe
(1/2 pt) dérivée :
 tous les constructeurs de la classe de base
 le constructeur par copie seulement
 le destructeur de la classe de base
 l'opérateur = s'il est surchargé
 l'opérateur + s'il est surchargé
 aucun des choix ci-dessus

Q6: Soit une **méthode val** d'une classe **Cessai**
(1/2 pt) déclarée ainsi : `Cessai val (int x);`
Que faut-il écrire dans la définition de cette méthode, pour retourner le résultat :
 `return this ;`
 `return *this ;`
 `return &this ;`
 ce n'est pas possible

Q7: Soit une classe C déclarée **amie** de la classe B, et
(1/2 pt) B est déclarée **amie** de la classe A :
 C accède à la partie privée de B
 B accède à la partie privée de C
 C accède à la partie privée de A
 A accède à la partie privée de C
 aucune proposition n'est exacte

Q8: Soit la classe D. Soit la classe E **dérivée** de D.
(1/2 pt) Soient les déclarations : `D * d1; E * e1;`
On peut écrire :
 `d1 = e1 ;`
 `e1 = d1 ;`
 `d1 = (D *) e1 ;`
 `e1 = (E *) d1 ;`
 aucune affectation n'est exacte

Q9: Quelles relations entre classes (ou entre objets) connaissez-vous ? Commentez.
(1 pt)

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Q10: Soit une classe Parent ; soit la classe Enfant dérivée de la classe Parent ; soit la classe PetitEnfant dérivée de la classe Enfant. Lorsqu'on écrit : `PetitEnfant p ;` quel constructeur de la classe PetitEnfant est appelé ? Qu'en est-il des constructeurs de la classe Enfant et des constructeurs de la classe Parent ?
 Détaillez précisément l'ordre d'appel des constructeurs.

.....

.....

.....

.....

.....

Q11: Que signifie en C++, l'écriture : `Vecteur * v = new Vecteur;` Que se passe-t-il ?
 (1 pt)

.....

.....

.....

.....

.....

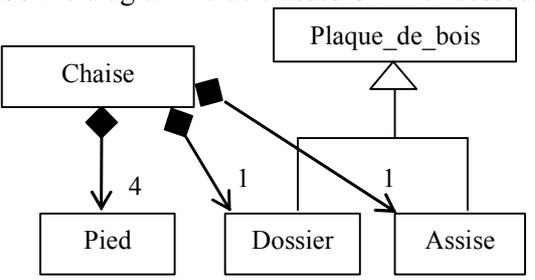
Q12: L'analyse d'une application pour un collège a permis d'isoler les entités suivantes:
 (1 pt)

- Collège
- Personnel
- Directeur
- Professeur
- Classe
- Elève

En utilisant la notation UML, faites un **diagramme de classe** en faisant apparaître les relations qui vous semblent pertinentes.

NB: on ne demande pas de définir les attributs ou les méthodes.

Q13: Soit le diagramme de classes UML ci-dessous :
 (1 pt)



Ecrivez un code C++ qui montre une mise en œuvre possible de ces classes et de leurs relations.

.....

.....

.....

.....

.....

.....

.....

Q14: Expliquez ce que fait l'instruction: `double * x = new double [9];`
 (1 pt)

.....

.....

Expliquez ce que fait l'instruction: `double * x = new double (9);`

.....

.....

.....

Q15: Soit la déclaration des classes A et B :

(1 pt)

```

class B ;
class A
{ int x;
public:
  int ajout(A t, B u)
  { x = t.x + u.y ;
    return x ; }
  ...
};

```

```

class B
{ int y;
public:
  int soustrait(A t, B u)
  { y = t.x - u.y ;
    return y ; }
  ...
};

```

Le code ci-dessus est-il
correct ?
Si non, encerclez la ou les
fautes dans le code source et
expliquez.
Donnez la correction.

Q16: Soit la déclaration de la classe Calpha :

(1 pt)

```

class Calpha
{ int x,y;
  Calpha (const Calpha &);
public:
  Calpha ();
  Calpha (int, int);
  ...
};

```

Et soit l'extrait de code suivant :

```

int main ()
{
  ...
  Calpha m1, m2(3,4), m3;
  Calpha m4 = m2;
  m3 = m2;
  ...
}

```

Le code ci-dessus est-il
correct ?
Si non, encerclez la ou les
fautes dans le code source et
expliquez.
Donnez la correction.

Q17: Le constructeur par copie d'une classe Engin est déclaré par : Engin(const Engin &);. Pourquoi ne peut-on pas écrire plus simplement cette déclaration : Engin(Engin); ? Justifiez votre réponse.

(1 pt)

.....
.....
.....
.....

Q18: Le code ci-dessous est-il correct ? Si non, encerclez la ou les fautes et expliquez. Donnez la correction.

(1 pt)

```

class Basel
{ int x;
public:
  Basel(int xx) { x = xx; }
};
class F1 : public Basel
{ int y;
public:
  F1(int xx, int yy)
  { Basel(xx); y = yy ; }
};

```

.....
.....
.....
.....
.....
.....
.....

Q19: Qu'est-ce qu'un attribut de classe ? Comment le note-t-on dans un diagramme de classes en UML ?

(1 pt)

.....
.....
.....

Et comment le déclare-t-on en C++ ?

.....
.....

Q20: Qu'est-ce que la STL ?
(1 pt)

.....
.....
.....
.....
.....

Q21: Implémenter les définitions des 3 méthodes de la classe Train déclarée comme suit :

```
(2 pt) class Train  
{  
private :  
    Loco l;  
    int nbwagon;  
    Wagon * w;  
public:  
    Train (Loco, int = 0,  
           Wagon * = NULL);  
    Train (const Train &);  
    Train operator+ (Train);  
    ...  
};
```

On suppose que les classes Loco et Wagon sont définies et qu'elles surchargent l'opérateur =.

Pour être complet, quelles autres méthodes faudrait-il surcharger dans la classe Train étant donnée sa mise en œuvre ?

.....
.....
.....

Q22: Le code ci-dessous est-il correct ? Si non, encerclez la ou les fautes et expliquez. Donnez la correction.

```
(1 pt) #include <string>  
using namespace std;  
  
string & fonc (int a)  
{  
    string s;  
    ...  
    return s;  
}
```

.....
.....
.....
.....
.....
.....

Q23: Soit la séquence de code suivante. Réécrivez le code pour rendre la classe Mat2x2 générique de telle manière que l'attribut val puisse être de tout type T défini par l'utilisateur.

```
(1 pt) #include <iostream>  
using namespace std;  
  
class Mat2x2  
{  
    int val[2][2];  
public:  
    Mat2x2 (int a, int b,  
           int c, int d);  
    int & elem (int i, int j)  
        { return val[i][j]; }  
    friend ostream & operator<<  
        (ostream &, const Mat2x2 &);  
};
```

.....
.....
.....
.....
.....
.....
.....
.....
.....