

NOM: \_\_\_\_\_ Prénom: \_\_\_\_\_ Groupe TD: \_\_\_\_\_

**Examen d'informatique 2<sup>ème</sup> Année**  
**Ecole Centrale de Lyon - département MI**  
(Mercredi 5 décembre 2001 - durée 2h00)

Chaque question est notée sur 1 point. Toutes les questions sont indépendantes entre elles. 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.**

**Q1:** On peut considérer l'envoi d'un message électronique comme la transmission d'un fichier texte. Quelles différences et similitudes faites-vous avec la transmission du même fichier par FTP ?

---

---

---

---

---

**Q2:** Dans les salles des stations des bâtiment F7 et H10, toutes les machines sont configurées comme des terminaux X-window. Si vous travaillez sur un de ces terminaux et que vous lancez un programme où s'exécute-t-il ? Quelles sont les conséquences sur les performances si la salle est pleine d'étudiants ?

---

---

---

---

---

**Q3:** L'analyse d'une application "Jeu de dames" a permis d'isoler les entités suivantes:

- Partie\_de\_Dames
- Echiquier
- Joueur
- Case
- Pion
- Joueur\_humain
- Joueur\_machine

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

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

**Q4:** Que met-on dans un fichier en-tête XXX.h ? A quoi sert ce fichier ?

---

---

---

---

**Q5:** A quoi sert *typedef* ? Donnez un exemple.

---

---

---

---

**Q6:** Soit une classe *Exemple*. Ecrivez une **méthode** (que l'on nommera *echange*) pour cette classe qui échange la valeur de ses 2 arguments entiers. Donnez le prototype et la définition de la méthode.

---

---

---

---

---

---

**Q7:** Un constructeur peut-il avoir des arguments avec valeur par défaut ? Si oui, donnez un exemple.

---

---

---

**Q8:** Quel est l'usage principal d'un destructeur ?

---

---

---

Un destructeur peut-il avoir des paramètres ? \_\_\_\_\_

Un destructeur peut-il être déclaré virtuel ? \_\_\_\_\_

**Q9:** Cette séquence est-elle correcte? Sinon encerclez la faute et corrigez-la.

```
class Machin
{ private:
  int a;
  float * b;
public :
  Machin()
  { a = 100;
    b = new float[50]; }
  ~Machin()
  { delete [] a;
    delete [] b; }
};
```

---

---

---

---

---

---

**Q10:** Dans quel cas est-il important de définir un constructeur par copie ? Donnez un exemple simple.

---

---

---

---

---

**Q11:** Pourquoi, dans un constructeur par copie, l'argument (qui est une instance de la classe elle-même) doit être passé par référence et non par valeur ?

---

---

---

---

**Q12:** A propos de la surcharge d'opérateur, laquelle de ces affirmations est **fausse** :

- le nombre d'opérandes acceptées par un opérateur ne peut être changé
- l'opérateur ternaire ( ? : ) ne peut être surchargé
- la surcharge d'opérateur est une forme de surcharge de fonction ou de surcharge de méthode
- la sémantique d'un opérateur ne peut être changée (par exemple, l'opérateur + doit toujours faire la somme de ses opérandes) même si le type des données auquel il s'applique peut être changé.

**Q13:** Soit la classe Point3D définie ci-dessous. Surchargez l'opérateur == afin de permettre de tester la coïncidence de 2 points sous la forme d'un fonction membre (méthode). Donnez le prototype et la définition de la méthode.

```
class Point3D
{
private:
    double x, y, z;
public:
    Point3D ( double c1=0.0,
             double c2=0.0,
             double c3=0.0 )
        { x=c1; y=c2; z=c3; }
    ...
};
```

---

---

---

---

---

---

---

---

---

---

**Q14:** Soit une classe Matrice supposée définie et soit le code suivant :

```
Matrice a, b, c;
a = b * c;
```

Ce code est-il correct si l'opérateur \* n'est pas surchargé ?

De même, ce code est-il correct si l'opérateur = n'est pas surchargé ?

Expliquez.

---

---

---

---

---

---

---

---

---

---

- Q15:** Une fonction "friend" d'une classe a-t-elle accès à la section privée de cette classe ?  oui  non  
Une fonction "friend" d'une classe a-t-elle accès à la section protégée de cette classe ?  oui  non  
Une fonction "friend" d'une classe a-t-elle accès à la section publique de cette classe ?  oui  non

**Q16:** Qu'est-ce qu'une méthode virtuelle ? A quoi sert-elle ?

---

---

---

---

**Q17:** Qu'est-ce que *this* ?

---

---

---

---

**Q18:** La séquence de code suivante est-elle correcte ? Si non, pourquoi ?

```
int taille = 64;  
float * truc = new float[taille];
```

---

---

La séquence de code suivante est-elle correcte ? Si non, pourquoi ?

```
int taille = 64;  
float truc [taille];
```

---

---

**Q19:** Cochez **toutes** les affirmations qui sont **vraies** :

- En C++, il est possible d'avoir des tableaux où les éléments du tableau sont des données de types différents.
- En C++, il est possible d'avoir des listes chaînées où les éléments de la liste sont des pointeurs vers des données de types différents.
- En C++, il est possible d'avoir des listes chaînées où les éléments de la liste sont des données de types différents.
- En C++, il est possible d'avoir des tableaux où les éléments du tableau sont des pointeurs vers des données de types différents.

**Q20:** Soit la séquence de code suivante. Comment qualifie-t-on la classe Bidule ? Pour que ce code soit correct, quels sont les prérequis sur la classe paramètre T ?

```
#include <iostream.h>  
template <class T> class Bidule  
{  
private:  
    T num1;  
public:  
    Bidule () {}  
    Bidule (T n) : num1 (n) {}  
    void print()  
        { cout << num1 << endl; }  
};
```

---

---

---

---

---

---

---

---