

NOM:  Prénom:  Groupe TD:

**Examen d'informatique 2<sup>ème</sup> Année**  
**Ecole Centrale de Lyon - département MI**  
(mardi 11 janvier 2005 - durée 1h30)

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:** La Société « Guillaume Portail » a développé une bibliothèque de classes composée des deux (1 pt) fichiers `fenetres.h` et `fenetres.cc`. Pour que cette bibliothèque soit vendue et utilisable par les clients sans que le code source soit dévoilé, quels fichiers doivent être distribués ? Expliquez votre réponse.

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

**Q2:** En C++, dans l'héritage entre deux classes, qu'est-ce qui est hérité et qu'est-ce qui ne l'est pas ? (1 pt)

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

**Q3:** Soit une classe C1 et soit la classe C2 dérivée de la classe C1 par héritage public. (1 pt) L'écriture : `C1 * obj1 = new C2;` est-elle valide ? Justifiez votre réponse.

.....  
.....

L'écriture : `C2 * obj1 = new C1;` est-elle valide ? Justifiez votre réponse.

.....  
.....

**Q4:** On souhaite concevoir un guichet (1 pt) automatique de banque (GAB). On a identifié les classes suivantes :

- GAB
- Carte\_bancaire
- Personne
- Client
- Exploitant
- Banque
- Receptacle
- Receptacle\_a\_billets
- Receptacle\_a\_recus

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).



**Q10:** Si la classe A est amie de la classe B est-ce que la classe B est aussi amie de A ?  oui  non (1 pt)

Si la classe A est amie de la classe B et la classe B est amie de la classe C, est-ce que la classe A est amie de la classe C ?  oui  non

**Q11:** Soit le diagramme de classes ci-contre et le code de la méthode afficher() de la classe C. (2 pt)

Lister tous les attributs de la classe B1.

.....  
 .....

Lister tous les attributs de la classe C.

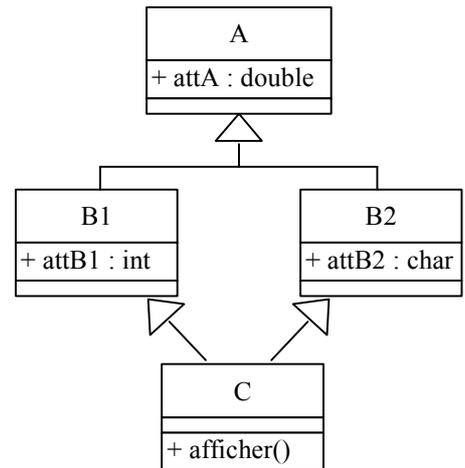
.....  
 .....

Quel est le problème posé par l'implémentation de la méthode afficher() de la classe C ?

.....  
 .....

Proposez une solution pour résoudre ce problème.

.....  
 .....



```

void C::afficher () {
    cout << attA;
    cout << attB1;
    cout << attB2;
}
  
```

**Q12:** Le code ci-dessous est-il correct ? Si non, encerclez la ou les fautes et expliquez. Donnez la correction. (1 pt)

```

class Essai
{
    ...
    public:
        int fonc (int);
        int fonc (char);
        double fonc (double) ;
};
  
```

.....  
 .....

**Q13:** Soit la classe Cercle définie ci-dessous. Surchargez l'opérateur == afin de permettre de tester la coïncidence de deux cercles sous la forme d'une **méthode inline**. (1 pt)

```

class Cercle
{ private :
    double centre_x, centre_y;
    double rayon;
    public:
        Cercle (double x, double y,
                double r) {
            centre_x = x ;
            centre_y = y ;
            rayon = r ; }
};
  
```

.....  
 .....

**Q14:** Soit la classe Date suivante :

```

(1 pt) class Date
{ int jour, mois ;
  char * mois ;
  public:
    Date & operator= (const Date &);
    ...
};
Date & Date::operator= (const Date & d) {
    if (this != &d) {
        delete [] mois;
        mois=new char[strlen(d.mois)+1)];
        strcpy (mois, d.mois);
        jour = d.jour; an = d.an;
        return *this ;
    }
}
  
```

A quoi sert le test « (this != &d) » dans le code de l'opérateur = ?

.....  
 .....

Que se passerait-il si on ne le faisait pas ?

.....  
 .....

**Q15:** Soit la classe Vecteur définit ci-dessous.

```
(1 pt) class Vecteur
{   int v[5];
    public:
        Vecteur (int x)
        { for(int i=0;i<5;i++) v[i]=x; }
        ...
};
int main () {
    Vecteur v1 (3);
    cout << v1[2];
    v1[3] = 8 ;
}
```

Ecrivez la surcharge de l'opérateur [] pour que le code de la fonction *main* soit juste.

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

**Q16:** Soient les 3 classes M1, C1 et F. Compléter la **définition inline** du constructeur de C1.

```
(1 pt) class M1
{   int m;
    public:
        M1 (int a) { m = a; }
};
class C1
{   char c;
    public:
        C1 (char x) { c = x; }
};
```

```
class F : public M1
{   C1 comp;
    double d;
    public:
        C1 (int a, char b, double c)
```

.....  
.....  
.....  
.....  
};

**Q17:** Soit les classes Vehicule et Camion :

```
(1 pt) #include <iostream>
using namespace std;

class Vehicule
{ ...
    public:
        virtual void affiche()
        { cout << "vehicule\n" ; }
};
class Camion : public Vehicule
{ ...
    public:
        void affiche()
        { cout << "camion\n" ; }
};
```

Soit le programme principal :

```
int main ()
{
    Vehicule v1, * v2;
    Camion * c1 = new Camion;
    c1->affiche();
    v1.affiche();
    v2 = c1;
    v2->affiche();
}
```

Que s'affiche-t-il à l'écran?

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

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

```
(1 pt) class Essai
{ ...
    public:
        virtual int getVal() = 0;
};
int main () {
    Essai * e1;
    Essai e2;
}
```

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

**Q19:** Soit la séquence de code suivante :

```
(1 pt) #include <iostream>
using namespace std;

template <class T> class Mat2x2
{   T val[2][2];
    public:
        Mat2x2 (T a, T b, T c, T d);
        T & operator[] (int i, int j)
        { return val[i][j]; }
        void affiche (int i, int j)
        { cout << val[i][j]; }
};
```

Comment qualifie-t-on la classe Mat2x2 ?

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

Pour que ce code soit correct, quels sont les prérequis sur la classe paramètre T ?

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