NOM: | **Prénom:** | **Groupe TD:**

Examen d'informatique 2ème Année
Ecole Centrale de Lyon - département MI

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: Quelles différences et similitudes faites-vous entre la récupération d'un fichier sur un serveur FTP anonyme et le même fichier sur un serveur Web par le protocole HTTP ?

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

Q2: Soit le fichier "essai.h" suivant :

```
#ifndef _ESSAI_H
#define _ESSAI_H
void truc (int, float);
class Bidule
  {
    ...
  };
Bidule var_glob;
#endif
```

Quel est le rôle des lignes commençant par le symbole # ? Expliquez le problème qui peut se poser si on ne les met pas.

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

Q3: Le fichier "essai.h" ci-dessus (Q2) contient-il un problème potentiel en cas d'inclusions multiples ? Si oui, lequel ? Expliquez ce qui se passe.

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

Q4: Dans un programme C++, on trouve les 2 lignes suivantes :

```
int vx;
calcul_de_base (& vx);
```

Quel est, à votre avis, la déclaration correcte de la fonction calcul_de_base :

- ❑ `void calcul_de_base (int);`
- ❑ `void calcul_de_base (int *);`
- ❑ `void calcul_de_base (int &);`

Q5: Un tableau C/C++ est-il un type nommé ? ❑ oui ❑ non

Une structure C/C++ est-il un type nommé ? ❑ oui ❑ non

Comment nommer un type qui ne l'est pas ? Donnez des exemples.

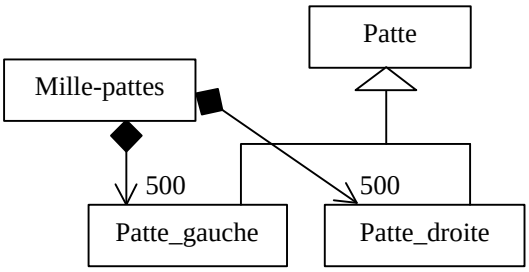
.....
.....
.....
.....
.....

Q6: L'analyse d'une application pour une société de location de véhicules a permis d'isoler les entités suivantes:
- Societe_de_location
- Vehicule
- Client
- Camion
- Voiture
- Moto

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

Q7: Soit le diagramme de classes UML ci-dessous :



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

Q8: Soit le petit programme :

```
int main ()
{
   Bidule y;
   cout << "Essai." << endl;
}
```

Comment concevoir la classe Bidule de façon à ce que le programme **affiche**, lors de l'exécution :

```
Creation d'un Bidule
Essai.
Destruction d'un Bidule
```

Q9: Si on a les classes suivantes :

```
classe Personne
{  char * nom;
 public:
   Personne (char *);
   Personne (const Personne &);
   ...
};
```

```
classe Entreprise
{  ...
 public:
   void embauche (Personne);
   ...
}
```

On considère la séquence de code suivante :

```
char * n = "Dupont";
Entreprise e;
e.embauche (n);
```

Que se passe-t-il ?

Q10: Soit le constructeur de la classe Cx :

```
Cx::Cx (int * a) : Cy (a)
{
   ...
}
```

Que peut-on dire sur les classes Cx et Cy ?

Q11: Dans quel cas un destructeur est-il **nécessaire** ?

Q12: A quoi sert un destructeur virtuel ? Donnez, un exemple mettant en œuvre 3 classes (Mere, Fille1 et Fille2).

Q13: Quelles sont les différences entre les 2 définitions, quels sont les avantages respectifs?

```
class Personne1
{   char nom[20];
  public :
    Personne1 (char *);
    ...
};
class Personne2
{   char * nom;
  public :
    Personne2 (char *);
    ...
};
```

Q14: Soit le code suivant :

```
class Simple
  {  int qui;
   public:
     Simple (int num) { qui = num;  cout << num << "-"; }
  };
Simple obj1_global (5);
Simple obj2_global (2);
int main ()
  {  Simple obj_local(3);  }
```

Quel est le résultat à l'écran de l'exécution de ce code ?

En modifiant la déclaration de obj_local de cette manière:

```
Simple obj_local (4, 3);
```

Le résultat sera-t-il modifié ? Justifiez votre réponse.

Q15: Soit une classe Truc dont le constructeur par copie est défini. Dans les différents cas ci-dessous, répondez oui si le constructeur par copie est appelé, non, s'il n'est pas appelé :

- Passage par valeur d'un paramètre du type Truc; exemple: void f1 (Truc t) { … } ❑ oui ❑ non
- Passage par référence d'un paramètre du type Truc; exemple: void f2 (Truc & t) { … } ❑ oui ❑ non
- Passage d'un paramètre du type pointeur sur Truc; exemple: void f3 (Truc * t) { … } ❑ oui ❑ non
- Retour de fonction d'une valeur de type Truc; exemple: Truc f4( ) { Truc t; … return t; } ❑ oui ❑ non

Remarque: pour chaque réponse juste: +0,25 point ; pour chaque réponse fausse: -0,25 point !

Q16 Soit les 2 classes : *(**Attention**: question double notée sur 2 points!)*

Q17

```
class Forme
 {  ...
  public:
    void qui_es_tu ()
     { cout<<"je suis une forme\n"; }
 };
```

```
class Cerle: public Forme
 {  ...
  public:
    void qui_es_tu ()
     { cout<<"je suis un Cercle, ";
       cout<<"et donc une Forme\n"; }
};
```

Commentez chacune des lignes de code suivantes, et indiquer quel est l'affichage (au cas où il y en a un) :

```
/* ligne 1 */
Cercle * ptrC1 = new Cercle();

/* ligne 2 */
ptrC1->qui_es_tu();

/* ligne 3 */
Forme * ptrFo = ptrC1;

/* ligne 4 */
ptrFo->qui_es_tu();

/* ligne 5 */
Cercle * ptrC2 = ptrFo;
```

Que faut-il modifier dans la déclaration des classes pour que la ligne 4 se comporte comme la ligne 2 ?

Q18: A quoi sert l'héritage multiple ? Donnez un exemple (vous pouvez l'illustrer par un diagramme UML).

Q19: Quel problème l'héritage multiple peut-il poser ? Donnez un exemple illustrant un tel problème.

Q20: Soit le code suivant :

```
class tab
{  int nb;
public:
   tab (int n=1) { nb=n; }
   friend ostream & operator<<
                    (ostream &, tab);
};
ostream & operator<<
             (ostream & os, tab t)
{ for (int i=0; i<t.nb; i++)
      {  os<<"\t";  }
  return os;
}
```

Quel se passe-t-il lorsque on écrit :

```
    cout << tab(2) << "essai";
```