

Aucun document ou support autre que le sujet ou les copies d'examen n'est autoris 
 (la copie ou les brouillons du voisin ne sont pas des supports autoris s).
 Positionnez imp rativement vos mobiles en mode « avion ».

1 Exercice

1. Si l'on dit que tous les chiens sont des animaux, qu'exprime-t-on ?
 - (a) Quel est le nom de cette relation ?
 - (b) Comment se repr sente-t-elle en UML (un exemple convaincant est bienvenu) ?
 - (c) Comment l'exprime-t-on en C++ ?
 - (d) Comment se nomme la technique employ e pour l'exprimer ?
2. Quelle(s) diff rence(s) y a-t-il entre g n ralisation et h ritage ?
3. La g n ralisation multiple est-elle un probl me ? Justifiez rapidement.
4. Quel probl me l'h ritage multiple pose t-il en C++ ?
5. Qu'est qu'une composition ?
 - (a) Repr sentez cette relation en UML   l'aide d'un exemple convaincant.
 - (b) Comment l'exprime t-on en C++ ?

2 Exercice

```

1 #include <iostream>
2 using namespace std;
3
4 class A {
5 public:
6   void f(int n) {
7     cout << "A::f" << endl;
8     if (n==0) return;
9     g(n-1);
10  }
11  void f() { cout << "A::f()" << endl; }
12  virtual void g(int n) {
13    cout << "A::g" << endl;
14    if (n==0) return; else g(n-1);
15  }
16 };
17
18 class B : public A {
19 public:
20   void f() { cout << "B::f()" << endl; }
21   virtual void g(int n) {
22     cout << "A::g" << endl;
23     if (n==0) return; else g(n-1);
24   }
25 };
26
27 int main() {
28   A a, *pa; B b, *pb;
29   pa = &a; a.f(); pa->f();
30   pb = &b; b.f(); pb->f();
31   a = b; a.f();
32   pa = pb; pa->f();
33   pa = new B; a.f(3);
34   exit(0);
35 }

```

1. À quoi sert l'instruction `using namespace std;` ?
2. Quels sont les affichages produit à l'exécution du programme ?
3. Que se passe-t-il si on qualifie la méthode `f()` de `virtual` ?
4. Quel est le nom de ce mécanisme ?

3 Exercice

1. Comment créer en C++ une classe abstraite ?
2. Comment rendre abstraite en C++ une classe entièrement définie (toutes les méthodes sont définies) ?
3. Qu'est-ce qu'une fonction amie ? Donnez un exemple.

4 Exercice

1. Créez en C++ un type `Array` qui permettra de contenir `N` entiers, où `N` est fixé à l'instanciation du type. (Exemple : `Array<10> a;`)
2. Ajoutez l'opérateur `[]` d'accès aux éléments et qui lève une exception en cas de débordement d'indice.
3. Ajoutez un opérateur `+` permettant de faire la concaténation de deux `Array`, la concaténation étant simplement la création d'un nouvel élément de type `Array` contenant d'abord les éléments du premier argument suivis par ceux du second.
4. Ajoutez un opérateur d'affichage pour le type `Array`.
5. Rendez le type générique.