



Laboratoire de langage C++ Interrogation n° 1

/50

R. Absil (abs)

7 Décembre 2015

Consignes générales

1. Notez votre nom, prénom, groupe et matricule sur chacune des feuilles que vous remettez à votre maître-assistant, y compris les brouillons.
2. Rédigez vos réponses en français correct : soignez votre orthographe et votre grammaire.
3. Justifiez chacune de vos réponses de façon complète et concise : une réponse incorrecte justifiée avec de bons arguments peut rapporter quelques points, alors qu'une bonne réponse non justifiée sera systématiquement comptée comme nulle.
4. L'interrogation est à cahier fermé.
5. L'interrogation dure 2h.

Rappel : le langage de référence est le C++11 (ISO/IEC 14882:2011) limité à son support par gcc 4.8.0

Question 1. Analysez les codes suivants, sachant qu'il n'y a pas d'erreur d'inclusion de fichier en-tête, et que toutes les options de compilation nécessaires sont fournies (p. ex., `-std=c++11`, `-fpermissive`, etc.). Pour chacun d'entre eux, indiquez

- s'il y a une erreur de compilation, à quelle(s) ligne(s) le cas échéant, et pourquoi;
- s'il y a une erreur lors de l'exécution, à quelle(s) ligne(s) le cas échéant, et pourquoi;
- si le programme s'exécute sans erreurs. Le cas échéant, indiquez les affichages qu'il génère, en spécifiant bien s'ils dépendent de l'architecture ou s'ils sont indéterminés.

Chaque code vaut 2 points. Répondez dans le tableau prévu à cet effet, situé à la page 7.

/28

Nom :

Prénom :

Groupe :

Matricule :

Code 1

```
1 #include <iostream>
2 int f(const char tab[]) {return sizeof(tab) / sizeof(*tab); }
3 int main()
4 {
5     char bat[] = {1, 2, 3, 4, 5, 6, 7};
6     std::cout << (sizeof(bat) / sizeof(*bat)) << std::endl;
7     std::cout << f(bat) << std::endl;
8 }
```

Code 2

```
1 #include <iostream>
2 using namespace std;
3
4 class A
5 {
6     public:
7     void f(int n) {cout << "A::entier_" << n << endl; }
8     void f(char n) {cout << "A::char_" << n << endl; }
9 };
10
11 class B : public A
12 {
13     public:
14     void f(int n, int m) { cout << "B::entiers_" << n << "_" << m << endl; }
15 };
16
17
18 int main()
19 {
20     int n = 1;
21     char c = 'a';
22     B b;
23     b.f(n);
24     b.f(c);
25 }
```

Code 3

```
1 #include <iostream>
2 #include <utility>
3
4 int main()
5 {
6     std::pair<char, double> pcd('Z', -3.14);
7     std::pair<float, long> pfl = pcd;
8     std::cout << pcd.first << "_" << pfl.second << std::endl;
9 }
```

Code 4

```
1 #include <iostream>
2 int f(const int tab[], int u) { return *(tab + u); }
3 int main()
4 {
5     int bat[] = {1,2,3,4,5,6,7};
6     std::cout << *(bat + 1) << std::endl;
7     std::cout << f(bat + 4, -2) << std::endl;
8 }
```

Nom :

Prénom :

Groupe :

Matricule :

Code 5

```
1 #include <iostream>
2
3 class Ball {};
4 class P
5 {
6     public:
7         P* target;
8         P(P* target) : target(target) {}
9         void aim(Ball b)
10        {
11            try
12            {
13                throw b;
14            }
15            catch(Ball& b)
16            {
17                std::cout << "Got_it_!" << std::endl;
18                target->aim(b);
19            }
20        }
21 };
22
23 int main()
24 {
25     P * parent = new P(0);
26     P * child = new P(parent);
27     parent->target = child;
28     parent->aim(Ball());
29 }
```

Code 6

```
1 #include <iostream>
2 int& f(int& i)
3 {
4     i *= 2;
5     return i;
6 }
7
8 int main()
9 {
10     auto i = 23;
11     std::cout << f(i)++ << std::endl;
12     f(i) = 321.123;
13     std::cout << i << std::endl;
14 }
```

Code 7

```
1 #include <iostream>
2 #include <set>
3
4 int main()
5 {
6     std::multiset<int> msi = {3, 5, 3, 6, 2};
7     for(auto e : msi)
8         std::cout << e << " ";
9     std::cout << std::endl;
10 }
```

Nom :

Prénom :

Groupe :

Matricule :

Code 8

```
1 #include <iostream>
2 using namespace std;
3
4 class exceptA {};
5 class exceptB : public exceptA {};
6 class exceptC : public exceptB {};
7
8 void f() { throw exceptB(); }
9
10 int main()
11 {
12     try
13     {
14         f();
15     }
16     catch(exceptA& e)
17     {
18         cout << "I_caught_an_A" << endl;
19     }
20     catch(exceptB& e)
21     {
22         cout << "I_caught_a_B" << endl;
23     }
24     catch(exceptC& e)
25     {
26         cout << "I_caught_a_C" << endl;
27     }
28 }
```

Code 9

```
1 #include <iostream>
2 using namespace std;
3
4 class point{
5     protected:
6         int x, y;
7     public:
8         point(int a = 0, int b = 0) : x(a), y(b) {}
9         virtual void affiche()
10         { cout << "Je_suis_en_" << x << "_," << y << endl; }
11 };
12
13 class pointcol : public point {
14     short col;
15     public:
16         pointcol(int a = 0, int b = 0, int c = 0) : point(a,b), col(c) {}
17         void affiche() {
18             cout << "Je_suis_en_" << x << "_," << y << "_et_ma_couleur_"
19                 << "est_" << col << endl; }
20 };
21
22 int main()
23 {
24     point * p = new point(3,5); pointcol * pc = new pointcol(8,6,2);
25     p = pc;
26     p->affiche(); pc->affiche();
27     p = new point(3,5); pc = (pointcol*)p;
28     p->affiche(); pc->affiche();
29 }
```

Code 10

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main()
6 {
7     int t[] = {1,2,3,4};
8     vector<int> v(t, t);
9     cout << v.size() << "_" << v.capacity() << endl;
10    for(int i = 0; i <= 8; i++)
11    {
12        v.push_back(0);
13        cout << v.size() << "_" << v.capacity() << endl;
14    }
15 }
```

Code 11

```
1 #include <iostream>
2 using namespace std;
3
4 class A
5 {
6     public:
7     A() {cout << "+A" << endl; }
8     A(const A&) {cout << "cA" << endl; }
9     virtual ~A() {cout << "-A" << endl; }
10 };
11 class B : public A
12 {
13     public:
14     B() {cout << "+B" << endl; }
15     B(const B&) {cout << "cB" << endl; }
16     virtual ~B() {cout << "-B" << endl; }
17 };
18
19 void f(A)
20 {
21     cout << "f" << endl;
22 }
23
24 int main()
25 {
26     B b; f(b);
27 }
```

Code 12

```
1 #include <iostream>
2 using namespace std;
3 class A
4 {
5     public:
6     const int i;
7     A(int entier) {entier = i;}
8 };
9
10 int main()
11 {
12     A a(2.5); cout << a.i << endl;
13 }
```

Code 13

```
1 #include <iostream>
2 using namespace std;
3
4 class A
5 {
6     public:
7         virtual A& operator=(const A&) { cout << "Aff_A" << endl;}
8 };
9 class B : public A
10 {
11     public:
12         virtual B& operator=(const B&) { cout << "Aff_B" << endl;}
13 };
14
15 int main()
16 {
17     A * a1 = new A(); A * a2 = new A();
18     B * b1 = new B(); B * b2 = new B();
19     b1 = b2; a1 = b1; a1 = b2;
20     a1 = new A(); a2 = new A();
21     b1 = new B(); b2 = new B();
22     *b1 = *b2; *a1 = *b1; *a1 = *b2;
23 }
```

Code 14

```
1 #include <iostream>
2 using namespace std;
3
4 class A
5 {
6     public:
7         int a;
8         A() { cout << "Def_A" << endl; }
9         A(int a) : a(a) { cout << "+A" << endl; }
10 };
11
12 class B : public A
13 {
14     public:
15         int b;
16         B(int a = 2, int b = 1) : A(a), b(b) { cout << "+B" << endl; }
17 };
18
19 class C : public A
20 {
21     public:
22         int c;
23         C(int a = 3, int c = 2) : A(a), c(c) { cout << "+C" << endl; }
24 };
25
26 class D : public B, public C
27 {
28     public:
29         int d;
30         D(int a = 4, int b = 5, int c = 6, int d = 7) : B(b,c), C(c,d), d(a)
31         { cout << "+D" << endl; }
32 };
33
34 int main()
35 {
36     D d(1,2,3,4);
37 }
```

Nom :

Prénom :

Groupe :

Matricule :

Code 1	Erreur compilation		
	Erreur exécution		
	OK		
Code 2	Erreur compilation		
	Erreur exécution		
	OK		
Code 3	Erreur compilation		
	Erreur exécution		
	OK		
Code 4	Erreur compilation		
	Erreur exécution		
	OK		
Code 5	Erreur compilation		
	Erreur exécution		
	OK		
Code 6	Erreur compilation		
	Erreur exécution		
	OK		
Code 7	Erreur compilation		
	Erreur exécution		
	OK		
Code 8	Erreur compilation		
	Erreur exécution		
	OK		

Nom : _____ Prénom : _____ Groupe : _____ Matricule : _____

Code 8	Erreur compilation		
	Erreur exécution		
	OK		
Code 9	Erreur compilation		
	Erreur exécution		
	OK		
Code 10	Erreur compilation		
	Erreur exécution		
	OK		
Code 11	Erreur compilation		
	Erreur exécution		
	OK		
Code 12	Erreur compilation		
	Erreur exécution		
	OK		
Code 13	Erreur compilation		
	Erreur exécution		
	OK		
Code 14	Erreur compilation		
	Erreur exécution		
	OK		

Nom : Prénom : Groupe : Matricule :

Question 2. Quelle est la différence entre la ligature statique et la ligature dynamique des liens en C++ ?

/5

Question 3. Qu'est-ce que le polymorphisme ? Dans quels cas le polymorphisme s'applique-t-il en C++ ? Quels sont les mots-clés associés ? Détaillez votre réponse aussi bien pour des conversions implicites que pour des appels de fonctions membres.

/9

Question 4. Citez 8 concepts ou fonctionnalités du C++ qui ne sont pas présentes en C, hormis tous les concepts relatifs au paradigme orienté objet.

/4

Question 5. À quoi sert la dérivation virtuelle (`class A : public virtual A`) et comment fonctionne-t-elle ?

/5