HE2B ESI

Langage C++
Interrogation n° 2

/120

R. Absil (abs)

2 décembre 2016

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.

Question 1. On souhaite que les codes suivants produisent l'affichage souhaité. Si ce n'est pas le cas, corrigez chacun de ces codes pour qu'ils produisent sans erreur l'affichage souhaité. Dans tous les cas, *justifiez également* les raisons pour lesquelles l'affichage souhaité apparaît ou non. /90

Remarques :

- ces codes peuvent ne pas compiler ;
- ces codes ne comprennent que des erreurs sémantiques et / ou de logique (ne cherchez donc pas un « `;` » ou un « `std::` » manquant) ;
- il est possible qu'il soit impossible de produire l'affichage souhaité, si tel est le cas, justifiez pourquoi ;
- chaque code vaut dix points, il y a neuf codes ;
- répondez sur le papier ministre, vous pouvez référencer le code via les numéros de ligne.

Nom : Prénom : Groupe : Matricule :

Code 1

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  class A
6  {
7    public:
8      ~A() { cout << "Boom" << endl; }
9  };
10
11 class B
12 {
13   vector<A*> v;
14   public  :
15     B() : v(vector<A*>(4)) {}
16 };
17
18 void f() { B b; }
19
20 int main()
21 {
22   f();
23   cout << "I want you in my room" << endl << "(outdated song)" << endl;
24 }
```

Affichage souhaité du Code 1

```
1  Boom
2  Boom
3  Boom
4  Boom
5  I want you in my room
6  (outdated song)
```

Code 2

```
1  #include <iostream>
2
3  int main()
4  {
5    int i = 33;
6    int const & ri = i;
7    std::cout << ri << std::endl;
8    i++;
9    std::cout << ri << std::endl;
10 }
```

Affichage souhaité du Code 2

```
1  33
2  34
```

Nom : Prénom : Groupe : Matricule :

Code 3

```
1  #include <iostream>
2  using namespace std;
3
4  class A
5  {
6    public:
7      void f(int n) {cout << "A::integer " << n << endl; }
8      void f(char n) {cout << "A::character " << n << endl; }
9  };
10
11 class B : public A
12 {
13   public:
14     void f(int n, int m) { cout << "B::integers " << 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   b.f(n, c);
26 }
```

Affichage souhaité du Code 3

```
1  A::integer
2  A::character
3  B::integers
```

Code 4

```
1  #include <iostream>
2  int f() {
3    static int i { 4 };
4    return --i;
5  }
6  int main() {
7    std::cout << f() << std::endl;
8    while (f()) { }
9    std::cout << f() << std::endl;
10 }
```

Affichage souhaité du Code 4

```
1  4
2  -1
```

Nom : Prénom : Groupe : Matricule :

Code 5

```
1  #include <iostream>
2
3  using namespace std;
4
5  class Integer
6  {
7    unsigned i;
8    bool positive;
9
10   public:
11     Integer(unsigned i, bool positive = true) : i(i), positive(positive) {}
12     Integer operator +(Integer a)
13     {
14       if(positive && a.positive) return Integer(i + a.i);
15       else if(positive && !a.positive) return Integer(i - a.i);
16       else if(!positive && a.positive) return Integer(a.i - i);
17       else return Integer(-i - a.i);
18     }
19
20     void print()
21     {
22       if(!positive)
23         cout << "-";
24       cout << i << endl;
25     }
26 };
27
28 int main()
29 {
30   Integer a1(2u);
31   Integer a2(3u,false);
32
33   Integer s = a1 + a2;
34   s.print();
35 }
```

Affichage souhaité du Code 5

```
1  -1
```

Nom : Prénom : Groupe : Matricule :

Code 6

```
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   catch(...)
29   {
30     cout << "I caught something" << endl;
31   }
32 }
```

Affichage souhaité du Code 6

```
1  I caught a B
```

Nom : Prénom : Groupe : Matricule :

Code 7

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  class A
7  {
8    public:
9      int i;
10     A(int i) : i(i) {}
11     void print() { cout << i << "␣"; }
12 };
13
14 int main()
15 {
16   vector<A> v(5);
17   for(int j = 0; j < v.size(); j++)
18     v[j].i = j;
19   for(A a : v)
20     a.print();
21 }
```

Affichage souhaité du Code 7

```
1  0 1 2 3 4
```

Code 8

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  using namespace std;
5  class A {
6    public:
7      int i;
8  };
9
10 int main()
11 {
12   vector<A> v = {5,4,3,2,1};
13   sort(v.begin(), v.end());
14   for(A a : v)
15     cout << a.i << "␣";
16 }
```

Affichage souhaité du Code 8

```
1  1 2 3 4 5
```

Nom : Prénom : Groupe : Matricule :

Code 9

```
1  #include <iostream>
2  using namespace std;
3  class A {
4    public:
5      A() { cout << "+A" << endl; }
6      A(const A& a) { cout << "rA" << endl; }
7      virtual ~A() { cout << "-A" << endl; }
8  };
9  class B {
10   public:
11     B() { cout << "+B" << endl; }
12     B(const B& b) { cout << "rB" << endl; }
13     virtual ~B() { cout << "-B" << endl; }
14 };
15 class C : public A, public virtual B {
16   public:
17     C() { cout << "+C" << endl; }
18     C(const C& c) { cout << "rC" << endl; }
19     virtual ~C() { cout << "-C" << endl; }
20 };
21
22 void f(A a) {}
23
24 int main() {
25   C c; cout << endl;
26   f(c); cout << endl;
27   C * cc = new C(); cout << endl;
28   delete cc; cout << endl;
29 }
```

Affichage souhaité du Code 9

```
1  +B   rA   +B   -C   -C
2  +A   -A   +A   -A   -A
3  +C        +C   -B   -B
```

Remarque 1. Les affichages ci-dessus sont à lire en colonnes, de haut en bas et de gauche à droite.

Question 2. Énoncez 10 concepts introduits par le standard `C++` qui ne sont pas présents dans le standard `C` et qui ne sont pas intrinsèquement liés au paradigme orienté objet. /5

Question 3. Expliquez la différence entre la ligature statique et la ligature dynamique des liens. /7

Nom : Prénom : Groupe : Matricule :

Question 4. Décrivez les différentes classes d'allocations de variables en `C++`, en quoi elles se différencient d'un point de vue gestion de la mémoire, comment les mettre en œuvre (mots clés associés), les mécanismes que certaines de ces classes rendent possibles, etc. /10

Question 5. Expliquez les différences entre pointeurs et références, entre autres d'un point de vue utilisation que d'un point de vue gestion de la mémoire. /8