heb esi

Langage C++
Interrogation n° 1

/110

R. Absil (abs)

17 octobre 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. /80

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 huit 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  class A{
3    int * t;
4    int n;
5    public:
6      A(int n) : t(new int[n]), n(n) {}
7      ~A() { delete t; }
8      void print()
9      {
10       for(int i = 0; i < n; i++)
11         std::cout << t[i] << " ";
12     }
13 };
14
15 void f(A a) { std::cout << "GET TO THE CHOPPAAA !!!" << std::endl; }
16
17 int main()
18 {
19   A a(5);
20   f(a);
21   a.print(); std::cout << std::endl;
22
23   A b(4);
24   b = a;
25   b.print(); std::cout << std::endl;
26
27   b = b;
28   b.print(); std::cout << std::endl;
29 }
```

Affichage souhaité du Code 1

```
1 GET TO THE CHOPPAAA !!!
2 0 0 0 0 0
3 0 0 0 0 0
4 0 0 0 0 0
```

Nom : Prénom : Groupe : Matricule :

Code 2

FICHIER A.CPP

```cpp
#include <iostream>

class B;

class A
{
  public:
    B * b;
    A() : b(nullptr) {}
    void print()
    {
      std::cout << "A_with_B" << std::endl;
    }
};
```

FICHIER B.CPP

```cpp
#include <iostream>

#include "A.cpp"

class B
{
  public:
    A a;
    B() {}
    void print()
    {
      std::cout << "B_with_A" << std::endl;
    }
};
```

FICHIER ABMAIN.CPP

```cpp
#include <iostream>

using namespace std;

int main()
{
  B b;
  A a;
  a.b = &b;
  b.a = a;
  a.print();
  b.print();
}
```

Affichage souhaité du Code 2

```
A with B
B with A
```

Nom : Prénom : Groupe : Matricule :

Code 3

```
1  #include <iostream>
2  int f(const char tab[])
3  {
4    return sizeof(tab) / sizeof(*tab);
5  }
6
7  int main()
8  {
9    char bat[] = {1, 2, 3, 4, 5, 6, 7};
10   std::cout << (sizeof(bat) / sizeof(*bat)) << std::endl;
11   std::cout << f(bat) << std::endl;
12 }
```

Affichage souhaité du Code 3

```
1  7
2  7
```

Remarque 1. Le code ci-dessus est exécuté sur un processeur 64 bits.

Code 4

```
1  #include <iostream>
2  #using namespace std;
3
4  int main()
5  {
6    int i = 5; int j = 2; double d = 1.5; //vous ne pouvez pas modifier cette ligne
7    cout << (i / j + d) << endl;
8  }
```

Affichage souhaité du Code 4

```
1  4
```

Nom : Prénom : Groupe : Matricule :

Code 5

```
1  #include <iostream>
2  using namespace std;
3  class A {
4    public:
5      A() { cout << "+A "; }
6      A(const A& a) { cout << "rA "; }
7      ~A() { cout << "-A "; }
8      A& operator =(const A& a) { cout << "=A " << endl; return *this; }
9  };
10
11 void f(A a) {}
12 void g(A& a) {}
13
14 int main()
15 {
16   A a;
17   f(a);
18   A * aa = new A();
19   aa = &a;
20   g(*aa);
21 }
```

Affichage souhaité du Code 5

```
1  +A rA -A +A -A -A
```

Code 6

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6    int a1 = 2.1;
7    cout << a1 << " ";
8    int a2(2.1);
9    cout << a2 << " ";
10   int a3{2.1};
11   cout << a3 << endl;
12 }
```

Affichage souhaité du Code 6

2 2 2

Nom : Prénom : Groupe : Matricule :

Code 7

FICHIER A.CPP

```cpp
#ifndef A_H
#define A_H
#include <iostream>
#include "B.cpp"

class A
{
  public:
    B b;
    A() {}
    void print()
    {
      std::cout << "A with B" << std::endl;
    }
};
#endif
```

FICHIER B.CPP

```cpp
#ifndef B_H
#define B_H
#include <iostream>
#include "A.cpp"

class B
{
  public:
    A a;
    B() {}
    void print()
    {
      std::cout << "B with A" << std::endl;
    }
};
#endif
```

FICHIER ABMAIN.CPP

```cpp
#include <iostream>

using namespace std;

int main()
{
  B b;
  A a;
  a.b = b;
  b.a = a;
  a.print();
  b.print();
}
```

Affichage souhaité du Code 7

```
A with B
B with A
```

Nom : Prénom : Groupe : Matricule :

Code 8

```
1  #include <iostream>
2  using namespace std;
3
4  void swap(int * i, int * j);//vous ne pouvez pas modifier ce prototype
5
6  int main()
7  {
8    int i = 2; int j = 3;
9    swap(&i, &j);
10   cout << i << " " << j << endl;
11 }
12
13 void swap(int * i, int * j)
14 {
15   int * tmp = i;
16   i = j;
17   j = tmp;
18 }
```

Affichage souhaité du Code 8

```
1  3 2
```

Question 2. Expliquez en détail le mécanisme d'inférence de type (comment il est mis en œuvre, à quoi sert-il, les mots-clé associés). /5

Question 3. Qu'est-ce que le mécanisme de la liste d'initialisation et dans quels cas ne peut-on pas s'en passer ? /5

Question 4. Détaillez comment mettre en œuvre des conversions implicites définies par l'utilisateur dans les cas suivants : /8

- d'un type de base vers un autre type de base,
- d'un type de base vers une classe,
- d'une classe vers un type de base,
- d'une classe vers une autre classe.

Question 5. Expliquez sommairement le principe de surcharge d'opérateur en C++, avec les mots-clé associés. Détaillez les limites de ce mécanisme, c'est-à-dire ce qu'il n'est pas possible de faire avec ce mécanisme. /12