

Héritage [he]

Exercices de cours

Karine Zampieri, Stéphane Rivière

Unisciel  algoprog  Version 20 mai 2018

Table des matières

1 Appréhender le cours	2
1.1 Constructeurs	2

C++ - Exercices de cours (Solution)

1 Appréhender le cours

1.1 Constructeurs



Objectif

Cet exercice se focalise sur le comportement des constructeurs.



Classe Premiere @[Premiere.hpp]

On considère la classe de base suivante :

```
class Premiere
{
public:
    Premiere()
    {
        cout<<"Constr Premiere"<<endl;
    }
};
```

1.1.1 Problème 1



Programme C++ @[qzconstr1.cpp]

```
#include <iostream>
using namespace std;
#include "Premiere.hpp"

class Seconde : public Premiere
{
    typedef Premiere super;
public:
    Seconde(bool b)
        : super()
    {
        cout<<"Constr Seconde"<<endl;
    }
};

class Troisieme : public Premiere
{
    typedef Premiere super;
public:
    Troisieme(int k)
        : super()
    {
        cout<<"Constr Troisieme"<<endl;
    }
};

class Quatrieme : public Premiere
{
    typedef Premiere super;
public:
    Quatrieme(double d)
```

```
    : super()
    {
        cout<<"Constr Quatrieme"<<endl;
    }
};

int main()
{
    {
        Premiere a1;
        Seconde a2(true);
        Troisieme a3(15);
        Quatrieme a4(12.3);
    }
}
```



Dessinez les relations d'héritage entre classes.



Prédisez ce qu'affiche le programme.



Compilez et exécutez-le pour vérifier vos prédictions.



Y a-t-il un constructeur qui s'exécute sans afficher de message à la création de certains objets ?

1.1.2 Problème 2



Programme C++ @[qzconstr2.cpp]

```
#include <iostream>
using namespace std;
#include "Premiere.hpp"

class Cinquieme : public Premiere
{
    typedef Premiere super;
public:
    Cinquieme()
    : super()
    {
        cout << "Constr Cinquieme" << endl;
    }
};

class Sixieme : public Cinquieme
{
    typedef Cinquieme super;
public:
    Sixieme()
    : super()
    {
```

```
        cout<<"Constr Sixieme"<<endl;
    }
};

int main()
{
    {
        Cinqieme a1;
        Sixieme a2;
    }
}
```



Complétez les relations d'héritage entre classes.



Prédisez ce qu'affiche le programme.



Compilez et exécutez-le pour vérifier vos prédictions.

1.1.3 Problème 3



Programme C++ @[qzconstr3.cpp]

```
#include <iostream>
using namespace std;
#include "Premiere.hpp"

class Septieme : public Premiere
{
    typedef Premiere super;
public:
    Septieme(int i)
    : super()
    {
        cout<<"Constr Septieme"<<endl;
    }
};

class Huitieme : public Septieme
{
    typedef Septieme super;
public:
    Huitieme()
    : super(0)
    {
        cout<<"Constr Huitieme"<<endl;
    }
};

int main()
{
    {
        Septieme a1(5);
        Huitieme a2;
```

```
}  
}
```



Le programme provoque une erreur à la compilation.

Essayez de trouver l'erreur. Si vous n'y arrivez pas, essayez de compiler ce programme : le message d'erreur peut vous mettre sur la voie.

Orientation

La classe `Huitieme` est incorrecte.

1.1.4 Problème 4



Programme C++ @[qzconstr4.cpp]

```
#include <iostream>  
using namespace std;  
#include "Premiere.hpp"  
  
class Neuvieme : public Premiere  
{  
    typedef Premiere super;  
public:  
    Neuvieme(int i)  
    : super()  
    {  
        cout<<"Constr1 Neuvieme"<<endl;  
    }  
    Neuvieme(bool b)  
    : super()  
    {  
        cout<<"Constr2 Neuvieme"<<endl;  
    }  
};  
  
class Dixieme : public Neuvieme  
{  
    typedef Neuvieme super;  
public:  
    Dixieme(double d)  
    : super(true)  
    {  
        cout<<"Constr1 Dixieme"<<endl;  
    }  
    Dixieme(int x, bool y)  
    : super(x)  
    {  
        cout<<"Constr2 Dixieme"<<endl;  
    }  
};  
  
int main()  
{  
    {
```

```
Dixieme a1(10.5);  
Dixieme a2(45, true);  
}  
}
```



Prédisez ce qu'affiche le programme.



Compilez et exécutez-le pour vérifier vos prédictions.