

POO sous C++ - TP4

Objectifs

L'héritage Multiple, Les fonctions virtuelles

Exercice 1

Quels seront les résultats fournis par ce programme :

```
#include <iostream>
using namespace std ;
class A{
int n ;
float x ;
public :
A (int p = 2){
n = p ; x = 1 ;
cout << "** construction objet A : " << n << " " << x << "\n" ;
}
} ;
class B{
int n ;
float y ;
public :
B (float v = 0.0){
n = 1 ; y = v ;
cout << "** construction objet B : " << n << " " << y << "\n" ;
}
} ;
```

Exercice 2

Même question que précédemment, en remplaçant simplement l'en-tête du constructeur de C par :

```
C (int n1=1, int n2=2, int n3=3, float v=0.0) : B(v)
```

Exercice 3

Quels seront les résultats fournis par ce programme :

```
#include <iostream>
using namespace std ;
class A{
int na ;
public :
A (int nn=1){
na = nn ; cout << "**construction objet A " << na << "\n" ;
}
} ;

class B : public A{
float xb ;
public :
B (float xx=0.0){
```

```

xb = xx ; cout << "***construction objet B " << xb << "\n" ;
}
} ;
class C : public A{
int nc ;
public :
C (int nn= 2) : A (2*nn+1){
nc = nn ;
cout << "***construction objet C " << nc << "\n" ;
}
} ;
class D : public B, public C{
int nd ;
public :
D (int n1, int n2, float x) : C (n1), B (x){
nd = n2 ;
cout << "***construction objet D " << nd << "\n" ;
}
} ;
main(){
D d (10, 20, 5.0) ;
}

```

Exercice 4

Transformer le programme précédent, de manière qu'un objet de type D ne contienne qu'une seule fois les membres de A (qui se réduisent en fait à l'entier na). On s'arrangera pour que le constructeur de A soit appelé avec la valeur $2*nn+1$, dans laquelle nn désigne l'argument du constructeur de C.

Exercice 5

Quels résultats produira ce programme :

```

#include <iostream>
using namespace std ;
class point{ protected : // pour que x et y soient accessibles à pointcol
int x, y ;
public :
point (int abs=0, int ord=0) { x=abs ; y=ord ; }
virtual void affiche (){
cout << "Je suis un point \n" ;
cout << " mes coordonnées sont : " << x << " " << y << "\n" ;
}
} ;
class pointcol : public point{
short couleur ;
public :
pointcol (int abs=0, int ord=0, short cl=1) : point (abs, ord){
couleur = cl ;
}
void affiche (){
cout << "Je suis un point coloré \n" ;
cout << " mes coordonnées sont : " << x << " " << y ;
cout << " et ma couleur est : " << couleur << "\n" ;
}
} ;
main(){
point p(3,5) ; point * adp = &p ;

```

```

pointcol pc (8,6,2) ; pointcol * adpc = &pc ;
adp->affiche () ; adpc->affiche () ; // instructions 1
cout << "-----\n" ;
adp = adpc ;
adp->affiche () ; adpc->affiche () ; // instructions 2
}

```

Exercice 6

Quels résultats produira ce programme :

```

#include <iostream>
using namespace std ;
class point{
int x, y ;
public :
point (int abs=0, int ord=0) { x=abs ; y=ord ; }
virtual void identifie (){
cout << "Je suis un point \n" ;
}
void affiche (){
identifie () ;
cout << "Mes coordonnées sont : " << x << " " << y << "\n" ;
}
} ;
class pointcol : public point{
short couleur ;
public :
pointcol (int abs=0, int ord=0, int cl=1 ) : point (abs, ord){
couleur = cl ;
}
void identifie (){
cout << "Je suis un point coloré de couleur : " << couleur << "\n" ;
}
} ;
main(){
point p(3,4) ;
pointcol pc(5,9,5) ;
p.affiche () ;
pc.affiche () ;
cout << "-----\n" ;
point * adp = &p ;
pointcol * adpc = &pc ;
adp->affiche () ; adpc->affiche () ;
cout << "-----\n" ;
adp = adpc ;
adp->affiche () ; adpc->affiche () ;
}

```