



Exemples pour mieux comprendre

Exercice n° 1

```
class A {  
    public static void main (String [] args){  
        A a,b,c ;  
        a=new A() ;   b=new A() ;           c=b ;   a=b ;  
    }  
}
```

Combien d'instances de la classe A sont créées pendant l'exécution du code suivant ?
Combien en reste après le passage du "Garbage collector" ?

Deux (02) instances.

Une seule, l'instance "new A()" n'étant plus référencée, sera supprimée par le GC

Exercice n° 2

```
class D {  
    public static int x ;  
    public int y ;  
    public static travailler() {x++;}  
    public D() {x++; y-- ; }  
}
```

Qu'affichera le code suivant ?

```
D.travailler() ; D a=new D() ; D b=new D() ; a.travailler() ;  
System.out.println(b.x + " et " + b.y) ;
```

4 et -1

D.travailler() ; //incrémente x de 0 à 1

D a=new D() ; //incrémente x de 1 à 2 et décrémente y de a de 0 à -1

D b=new D() ; //incrémente x de 2 à 3 et décrémente y de b de 0 à -1

a.travailler() ; // incrémente x de 3 à 4

Exercice n° 3

```
class D {  
    public int x ;  
    public D() { x=3 ; } ;  
    public D( int a){this() ; x=x+a ; } ;  
    public D( int a, int b){this(b) ; x= x-a ; }  
}
```



Qu'affichera le code suivant ?

```
D a=new D(5,6) ;  
System.out.println(a.x) ;
```

**D a=new D(5,6) ; // appelle this(b) qui lui-même appelle this() d'où x=3,
puis x=x+6 d'où x=9, puis x=x-5 et donc a.x affiche 4**

Exercice n° 4

```
class C {  
    public static int i;  
    public int j;  
    public C() {i++; j=i; }  
}
```

Qu'affichera le code suivant ?

```
C x=new C(); C y=new C(); C z= x;  
System.out.println(z.i + " et " + z.j);
```

C x=new C(); // incrémente la variable de classe i de 0 à 1 et x.j=1

C x=new C(); // la variable de classe i de 1 à 2 et y.j=1

C z= x; // z reference la même instance que x

Affiche 2 et 1