

## JAVA : TDN°1

### Exercice1 : Priorités des opérateurs arithmétiques

Éliminer les parenthèses superflues dans les expressions suivantes (l'ordre des calculs devant rester le même) :

- $(a + b) - (2 * c)$  // expression 1
- $(2 * x) / (y * z)$  // expression 2
- $(x + 3) * (n \% p)$  // expression 3
- $(-a) / (- (b + c))$  // expression 4
- $(x / y) \% (-z)$  // expression 5
- $x / (y \% (-z))$  // expression 6

### Exercice2 : Conversions implicites

Soit ces déclarations :

```
byte b1 = 10, b2 = 20 ; short p = 200 ;  
int n = 500 ;  
long q = 100 ;  
float x = 2.5f ;  
double y = 5.25 ;
```

Donner le type et la valeur des expressions arithmétiques suivantes :

```
b1 + b2 // 1  
p + b1 // 2  
b1 * b2 // 3  
q + p * ( b1 + b2); // 4  
x + q * n // 5  
b1 * q / x // 6  
b1 * q * 2. / x // 7  
b1 * q * 2. f / x // 8
```

### Exercice 3 : Exceptions flottantes

Quels résultats fournit ce programme ?

```
public class Excep { public static void main (String args[])  
{ double x1 = 1e200,  
x2 = 1e210 ;  
double y, z ;  
y = x1 * x2 ;  
System.out.println (" valeur de y " + y) ;  
x2 = x1 ; z = y / ( x2 - x1) ;
```

ISET NABEUL

```

System.out.println (y + " divise par " + (x2-x1) + " = " + z) ;
y = 15 ;
z = y/( x2-x1) ;
System.out.println (y + " divise par " + (x2-x1) + " = " + z) ;
z = (x2-x1)/( x2-x1) ;
System.out.println (( x2-x1) + " divise par " + (x2-x1) + " = " + z) ;
System.out.println (z + " + 1 = " + (z + 1)) ;
x1 = Float.POSITIVE_INFINITY ;
x2 = Double.NEGATIVE_INFINITY ;
z = x1/ x2 ;
System.out.println (x1 + "/" + x2 + " = " + z) ; } }

```

**Exercice 4 : Type char**

Soit ces déclarations :

```

char c = 60,
ce = 'e',
cg = 'g' ;
byte b = 10 ;

```

Donner le type et la valeur des expressions suivantes :

```

c + 1
2 * c
cg - ce
b * c

```

**Exercice 5 : Opérateurs logiques**

Quels résultats fournit ce programme ?

```

public class CourCir {
    public static void main (String args[])
    {
        int i = 10,
        j = 5 ;
        if (i < 5 && j ++ < 10)
            System.out.println ("&& 1 vrai") ;
        else
            System.out.println ("&& 1 faux") ;
    }
}

```

```

        System.out.println (" i = " + i + " j = " + j) ;
        if (i < 5 & j ++ < 10)
            System.out.println ("& vrai") ;
        else
            System.out.println ("& faux") ;
        System.out.println (" i = " + i + " j = " + j) ;
        if (i < 15 && j ++ < 10)
            System.out.println("&& 2 vrai") ;
        else
            System.out.println ("&& 2 faux") ;
        System.out.println (" i = " + i + " j = " + j) ;
        if (i < 15 | | j ++ < 10)
            System.out.println (" | | vrai") ;
        else
            System.out.println (" | | faux") ;
        System.out.println (" i = " + i + " j = " + j) ;
    }
}

```

**Exercice6 : Priorités des opérateurs**

Éliminer les parenthèses superflues dans les expressions suivantes :

```

a = (x + 5) // 1
a = (x = y) + 2 // 2
a = (x = (y + 2)) // 3
(a < b) && (c < d) // 4
(i ++ ) * (n + p) // 5
x ++ = (n % p) // 6
n = (p ++ = 5) // 7

```

**Exercice7 : Affectation et conversion**

Soit ces déclarations :

```

byte b ;
short p ;
int n ;
long q ;
final int N = 10 ;

```

**float x ;**

**double y ;**

Parmi les expressions suivantes, lesquelles sont incorrectes et pourquoi ? Lorsque l'expression est correcte, citer les conversions éventuellement mises en jeu.

**b = n // 1**

**b = 25 // 2**

**b = 500 // 3**

**x = 2\* q // 4**

**y = b\* b // 5**

**p = b\* b // 6**

**b = b + 5 // 7**

**p = 5\* N-3 // 8**

### Exercice8 : Opérateurs d'incrément, de décrémentation et d'affectation élargie

1. Quels résultats fournit ce programme ?

**public class OpIncr**

**{**

**public static void main( String[] args)**

**{**

**int i, j, n ;**

**i = 0 ;**

**n = i ++ ;**

**System.out.println (" A : i = " + i + " n = " + n ) ;**

**i = 10 ;**

**n = ++ i ;**

**System.out.println (" B : i = " + i + " n = " + n ) ;**

**i = 20 ;**

**j = 5 ;**

**n = i ++ \* ++ j ;**

**System.out.println (" C : i = " + i + " j = " + j + " n = " + n ) ;**

**i = 15 ;**

**n = i + = 3 ;**

**System.out.println (" D : i = " + i + " n = " + n ) ; i = 3 ; j = 5 ; n = i \*= --j ;**

**System.out.println (" E : i = " + i + " j = " + j + " n = " + n ) ;**

**}**

**}**

2. Soit ces déclarations :

**byte b ;**

**short p ;**

**char c ;**

**int n ;**

**float x ;**

Parmi les expressions suivantes, lesquelles sont incorrectes et pourquoi ?

**c = c + 1 // 1**

**c ++ // 2**

**c + = 3 // 3**

**b + = c // 4**

**p + = b // 5**

**p = p + b // 6**

**n + = x // 7**

**n = n + x // 8**

**x ++ ; // 9**