

Prog 1 - Exercices

1 Expressions, priorités

1. Qu'affichent les instructions suivantes ?

```
System.out.println(2+2);
System.out.println(3<=4);
System.out.println("1"+2+3+" soleil");
System.out.println(81+42+" soleil");
```

Rappel: L'opérateur + est interprété comme l'addition ou la concaténation selon le type de ses opérands (arguments). Dans une expression, l'identification des opérands d'un opérateur nécessite de révéler le parenthésage, souvent implicite, selon les règles priorité et d'évaluation de gauche à droite.

2. Quelle est la valeur de la variable booléenne `b` (précédemment déclarée) après chacune des affectations suivantes ?

```
b = false && false || true;
b = false && false == false;
b = true || false == false;
```

3. La variable `x` de type `int` a été déclarée et initialisée (elle contient une valeur). Qu'affichent les morceaux de programmes suivants ?

```
x = x+1;                               | int y = x;
System.out.println(x==x+1);           | System.out.println(y==x);
```

Rappel: Les symboles = et == n'ont pas la même signification (ou sémantique).

4. Chacune des instructions conditionnelles suivantes peut être simplifiée : dans chaque cas trouver une affectation équivalente (les déclarations `int a, b`; et `boolean nul, produitNul`; et l'initialisation de `a` et `b` ont été préalablement faites).

<pre>if(a==0) nul = true; else nul = false;</pre>	<pre>if(a!=0) nul = false; else nul = true;</pre>	<pre>if(a==0) produitNul = true; else if(b==0) produitNul = true; else produitNul = false;</pre>
---	---	--

Rappel: L'instruction conditionnelle `if(a>=0) positif = true; else positif = false;` est équivalente à l'affectation `positif = (a>=0);`

2 Portée des déclarations

5. Les programmes suivants sont-ils corrects ? Si non, pourquoi ?

<pre>{ int a; } a=0; int b=a;</pre>	<pre>int a; { a=0; } int b=a;</pre>	<pre>{ int a=0; } int b=a;</pre>	<pre>int a; { int a=0; } int b=a;</pre>
---	---	--------------------------------------	---

<pre>int a=0; { int b=a; }</pre>	<pre>{ int a=0; } { int a=0; }</pre>	<pre>int a; { a=0; } { int b=a; }</pre>
--------------------------------------	--	---

Rappel: Les blocs interfèrent avec les déclarations de variables (portée) mais pas avec les affectations. La portée d'une variable est l'ensemble des instructions suivant sa déclaration, dans le plus petit bloc qui contient cette déclaration. Autrement dit, une variable déclarée dans un bloc n'existe pas en dehors de ce bloc. En revanche, une affectation ayant lieu dans un bloc modifie définitivement la valeur de la variable. Tout le corps de la méthode `main` est un bloc.