

DÉPARTEMENT D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE

SIGLE DU COURS: IFT 1010 (H001)

NOM DU PROFESSEUR: Philippe Langlais

TITRE DU COURS: Programmation I

EXAMEN INTRA

Date : 22 février
Heure : 8:30 - 10:30
Salle : 0030 / 1360

DIRECTIVES PÉDAGOGIQUES:

- Toute documentation est permise.
- **Inscrivez tout de suite votre nom et code permanent dans l'encadré.**
- Répondez **sur le questionnaire** en utilisant l'espace libre qui suit chaque question.

NOTATION:

Exercice 1:	/20	A)	B)	C)	D)	
Exercice 2:	/30	A)	B)	C)	D)	E)
Exercice 3:	/20					
Exercice 4:	/30					
Total:	/100					

Inscrivez votre nom et votre code permanent ici

CONSEIL:

Les deux premiers exercices contrôlent vos connaissances générales de Java. N'y passez pas un temps déraisonnable (vous connaissez la réponse ou pas). Les deux exercices qui suivent vérifient votre aptitude à la programmation structurée (prenez le temps de réfléchir).

Exercice 1 (20 points):

Dans cet exercice, il vous est demandé de trouver des erreurs de compilation et/ou des erreurs d'exécution. Les erreurs de compilation sont des erreurs qui rendent impossible la compilation (javac). Les erreurs logiques ou erreurs d'exécution sont des erreurs qui ne sont pas détectées à la compilation, mais qui se manifestent à l'exécution par un comportement non prévu.

Les exemples de code qui sont donnés ne sont pas mis en contexte (dans une classe particulière), ceci afin de réduire la taille de l'énoncé, les erreurs qu'on vous demande de trouver ne sont pas liées à ce fait.

Question A (5 points)

Ce code comporte une erreur qui empêche sa compilation. Expliquez clairement cette erreur.

```
boolean res;  
if (i > 4) res = true;  
else if (i<4) res = false;  
  
if (res) System.out.println(i + " est superieur a 4 ");  
else System.out.println(i + " n'est pas superieur a 4");
```

Question B (5 points)

Ce code contient une erreur logique. Expliquez l'erreur et corrigez-là directement dans le code.

```
int somme = 0;  
for (int n=0; n<=10; n++) if ( (n % 2) == 0) somme += n;  
System.out.println("1 + 3 + 5 + 7 + 9 = " + somme);
```

Question C (5 points)

Quel est l'affichage provoqué par l'exécution de ce code ?

```
int x= 6;
int y= 4;
if (x > 5)
    if (y>5) System.out.println("x et y sont supérieurs à 5");
else
    if (y<5) System.out.println("x et y sont inférieurs à 5");
```

Question D (5 points)

Quels affichages sont produits par les instructions suivantes ?

```
System.out.println(" 3 * 4 = " + 3 * 4);
System.out.println(" 3 + 4 = " + 3 + 4);
System.out.println(" A = " + (3 / 2 * 4 -2) );
System.out.println(" B = " + ((14<15)? (2 + 7/8 - 1 - 2): 4));
System.out.println(" C = " + 4 % 2 + 3 );
```

Exercice 2 (30 points):

Répondez à chaque question sachant les deux déclarations suivantes:

```
float f;  
int i;
```

Question A (5 points)

Rayez les instructions qui provoquent une erreur de compilation.

```
f = 3;  
f = 3.0;  
i = 3.5;  
  
i = 10 / 2.0;  
i = (int) 10 / 2.0;
```

Question B (5 points)

Quelle est la valeur de `i` après cette affectation ?

```
i = (int) Math.random() * 15 + 1;
```

Écrivez une instruction qui affecte à `i` une valeur entière entre 4 et 17 (bornes incluses):

Question C (5 points)

Indiquez à même le code (directement après l'instruction), l'affichage produit par ce code. Si une instruction contient une erreur de compilation ou une erreur d'exécution, rayez-là et indiquez à côté (de manière concise) la nature de l'erreur.

```
String s = "bonjour";  
System.out.println(s.charAt(0));  
System.out.println(s.length());  
System.out.println(s.charAt(s.length()));
```

Question D (10 points)

Indiquez l'affichage produit par ce code:

```
int y=5, x=4;  
System.out.println(y++);  
if ((x++ > 2) || (++y == 4)) System.out.println("A");  
else System.out.println("B");  
System.out.println("x=" + x + " y=" + y);
```

Question E (5 points)

La méthode suivante est susceptible de provoquer une erreur d'exécution. Indiquez pourquoi et proposez une correction qui ne change pas la sémantique du code.

```
void methode(int i) {  
    if ((i != 0) || ( (15/i) == 3)) System.out.println(i);  
}
```

Exercice 3 (20 points):

Quel est l'affichage produit par la méthode `exercice3` ?

```
void exercice3() {
    int c;
    final int MAX = 10;

    for (int l=0; l<=MAX/2; l++) {
        for (c=0; c<l; c++) System.out.print(".");
        System.out.print(l);
        while (c<= (MAX-1)) {
            if (c == (MAX/2)) System.out.print(l);
            else System.out.print(".");
            c++;
        }
        System.out.print(l);

        while (c <= MAX) {
            System.out.print(".");
            c++;
        }
        System.out.println();
    }
}
```



Votre réponse

Exercice 4 (30 points):

Écrivez un programme qui demande à un utilisateur de saisir deux chaînes de caractères (resp. `s1` et `s2`) et qui indique à l'utilisateur si la première chaîne (`s1`) est une sous-chaîne de la seconde (`s2`) ou pas. Si `s1` est une sous-chaîne de `s2`, alors votre programme doit également indiquer l'indice (dans `s2`) à partir duquel commence `s1`.¹

Voici trois exemples d'exécution de ce programme:

```
Entrez une chaîne: au
Entrez une chaîne: saumon
La chaîne "au" est une sous-chaîne de "saumon".
Elle commence à la position 1
```

```
Entrez une chaîne: auk
Entrez une chaîne: saumon
La chaîne "auk" n'est pas une sous-chaîne de "saumon".
```

```
Entrez une chaîne: mon
Entrez une chaîne: saumon
La chaîne "mon" est une sous-chaîne de "saumon".
Elle commence à la position 3
```

Note: Les seules méthodes de la classe *String* que vous devez/pouvez utiliser sont:

`char charAt(int i)` qui retourne le caractère d'indice `i` d'une chaîne
`int length()` qui retourne la longueur de la chaîne

En d'autres termes, n'utilisez pas de méthodes du genre `indexOf` c'est précisément ce qu'on vous demande de programmer !

¹Si `s1` est contenue plusieurs fois dans `s2`, alors l'indice retourné est l'indice de première apparition de `s1` dans `s2`.

