Algorithmes et Pensée Computationnelle

Programmation de base - Exercices de base

Le but de cette séance est d'aborder des notions de base en programmation. Au terme de cette séance, l'étudiant sera capable de :

- Définir une variable, définir son type et sa valeur.
- Définir une fonction et comprendre son rôle.
- Utiliser des notions d'algèbre booléenne.
- Comprendre la notion d'entrée/sortie.

Les langages qui seront utilisés pour cette séance sont Java et Python. Assurez-vous d'avoir bien installé Intellij. Si vous rencontrez des difficultés, n'hésitez pas à vous référer au guide suivant : tutoriel d'installation des outils et prise en main de l'environnement de travail.

1 Représentation de nombres entiers

Question 1: (**0** 5 minutes) **Entiers non signés** Sur 8 bits convertir $113_{(10)}$ en base binaire.



Faire un tableau comme présenté dans la diapositive 10 du cours de la semaine 3. Essayer de décomposer le nombre en une somme de puissances de 2.

Question 2: (*5 minutes*) **Entiers signés négatifs**

En utilisant le résultat de la question précédente, convertir sur 8 bits -113₍₁₀₎ en base binaire.

Conseil

- Il faut prendre la représentation sur 7 bits d'un nombre entier non signé.
- On rajoute un 8ème bit qui sera le signe.
- Pour cet exercice, reprendre l'expression non signée de la question précédente (question
- 1 Entiers non signés) et changer le premier bit en conséquence.
 Le premier bit vaut 0 pour un nombre positif et 1 pour un nombre négatif.

Question 3: (5 minutes) Complément à 1

Ecrire le complément à 1 de -113₍₁₀₎.

Quelle est la différence entre cette méthode et la précédente?

Conseil

- En programmation l'opposé d'une variable est **not**(la variable). Ici, le même principe s'applique. L'opposé de 0 en binaire est 1.
- Pour étudier la différence, il faut regarder les différentes manières d'exprimer -0 en binaire (se référer à la diapositive 11 du cours de la semaine 3).

Question 4: (5 minutes) Complément à 2

Quel est le complément à 2 de -113₍₁₀₎ et quelle est l'utilité de cette représentation?

Conseil

Exemple du cours avec $87_{(10)}$

 $87_{(10)} = 01010111_{(2)}$

a	0	1	0	1	0	1	1	1
b	1	0	1	0	1	0	0	0
С	1	0	1	0	1	0	0	1

a : convertir le nombre en binaire

b: inverser tous les bits

c : rajouter 1 au nombre pour obtenir le complément à 2

On obtient donc $-87_{(10)} = 10101001_{(2)}$

Question 5: (**1** *10 minutes*) **Floating point**

Voici la représentation en binaire d'un nombre à virgule flottante :

signe	exposant	mantisse
0	10110101	010000010000000000000000000000000000000

Que vaut cette représentation en base 10? Utiliser la représentation des floating point (avec un biais de 127) Arrondir les résultats intermédiaires et la valeur finale au 3ème chiffre significatif après la virgule.

- Se référer à la diapositive 15 du cours de la semaine 3 pour plus de détails.
- Poser chaque calcul, et ensuite tout fusionner avec la formule.
- L'exposant et la mantisse sont des entiers positifs, donc non signés.

2 Typage

Le but de cette partie des travaux pratiques est de comprendre les notions de typage statique et dynamique, ainsi que leur impact sur l'exécution et la gestion des erreurs.

Question 6: (**Q** 3 minutes) Les types de variables

Quelles sont les principaux types de variable (donnez en 3), comment les déclareriez vous en Python et en Java?



Se référer aux diapositives du cours ou à la documentation officielle de votre langage de programmation préféré.

Question 7: (5 minutes) Typage statique et dynamique

Parmi ces différents programmes, lesquels pourront être exécutés sans problème et lesquels lèveront des erreurs? Dans le cas où ils génèreraient des erreurs, expliquez la raison de ces erreurs.

Java:

```
// Programme 1
 1
 2
     int variable_1 = 0;
 3
     int variable 2 = 3;
     variable_2 = variable_1;
 4
 6
    // Programme 2
     var variable_1 = 0;
 7
 8
     var variable_2 = "Hello";
 9
     variable_2 = variable_1;
10
11
    // Programme 3
    int variable 1 = 0;
12.
     String variable_2 = "Hello";
13
     variable_2 = variable_1;
14
15
16
    // Programme 4
     var variable_1 = 0;
17
     variable_1 = 3.14;
18
19
    // Programme 5
20
21
     var variable_1 = 0;
     String variable_2 = "Hello";
    String variable_3 = "World";
23
24
     variable_2 = variable_3;
25
26
    // Programme 6
27
     int variable_1 = 0;
     variable_1 = 3.14;
     Python:
     # Programme 7
     variable_1 = 0
 2
     variable_2 = "Hello"
 4
     variable_2 = variable_1
 5
 6
     # Programme 8
 7
     variable_1 = 0
```

 $variable_{-}1 = 3.14$

Conseil

En Java, le type est statique, ce qui signifie qu'à partir du moment où vous créez une variable, un type va lui être attribué (par vous ou par le code en lui-même par déduction). Ce type ne pourra pas être changé, et si vous tentez de le faire (en lui attribuant une valeur d'un autre type par exemple), une erreur sera levée.

En Python, le typage est dynamique, il est donc tout à fait possible de changer le type d'une variable sans déclencher d'erreur.

A Erreurs fréquentes

Au cas où vous exécuteriez les programmes 2, 4 et 5 et rencontriez l'erreur suivante : Exception in thread "main" java.lang.UnsupportedClassVersionError: ... has been compiled by a more recent version of the Java Runtime (class file version 54.0), this version of the Java Runtime only recognizes class file versions up to 52.0, suivez les étapes ci-dessous :

- 1. Mettez à jour votre JDK en téléchargeant la version adéquate via le lien suivant : https://www.oracle.com/java/technologies/javase-jdk15-downloads.html.
- 2. Une fois votre JDK installé, redémarrez IntelliJ
- 3. Dans la barre de menus, cliquez sur **Run** > **Edit Configurations**.
- 4. Dans la fenêtre qui apparaitra (capture ci-dessous), sélectionnez la version du JRE la plus récente.

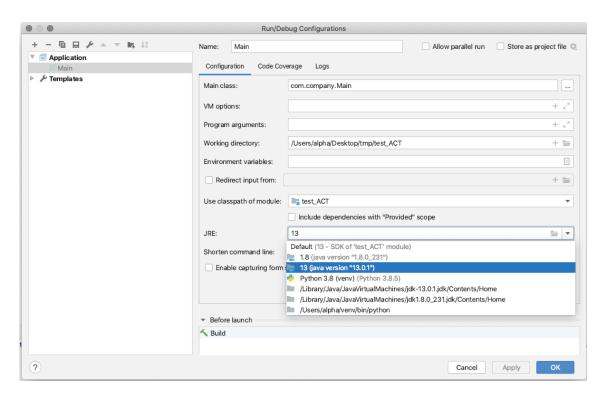


FIGURE 1 – Fenêtre de configuration de l'environnement de développement Java.

Question 8: (5 minutes) **Détection d'erreurs**

Soient les deux programmes suivants, l'un en Java et l'autre en Python. Chaque programme comporte une fonction nommée raise_error() levant une exception de type TypeError. Dans les deux cas, cette fonction ne sera pas appelée lors de l'exécution (la condition est remplie d'office). Pourtant, l'un de ces deux codes lèvera une erreur, et l'autre sera exécuté sans problème. Quel code lèvera l'erreur et lequel sera exécuté?

Justifiez votre réponse.

```
En Java:
     public class Main {
 2
        static void raise_error(){
 4 5
          int variable_1 = 0;
          variable_1 = "Bonjour";
 6
7
8
9
        public\ static\ void\ \frac{main}{string[]\ args)\ \big\{
10
            System.out.print("OK");
11
12
          else {
13
            raise_error();
14
15
16
     En Python:
     def raise_error():
 1
 2
        print("3" + 5)
 3
 4
     if True:
        print("OK")
 6
     else:
        raise\_error()
```

•

Conseil

Pensez à vérifier le type des variables. Faites attention à la différence principale entre le typage des variables en Java et le typage en Python.

3 **Bases en programmation**

Le but de cette section est d'écrire vos premières lignes de code. Les notions abordées concerneront les variables, les fonctions, et les interactions avec l'utilisateur (input/output). Vous pouvez les écrire en Java ou en Python.

Question 9: (5 minutes) **Output (Java ou Python)**

Créez une variable nom (str) contenant votre nom, et une autre prenom (str) contenant votre prénom puis affichez: "Bonjour, prenom nom".



Conseil

Utilisez la fonction print() de Python et System.out.println() de Java.

Ouestion 10: (5 minutes) **Input** (**Java ou Python**)

En vous référant à l'exercice précédent (Output (Java ou Python)), demandez à l'utilisateur d'entrer son nom et son prénom via la fonction input() au lieu d'initialiser vous-même les variables.



Conseil

Utilisez la fonction input() en Python, la classe Scanner() en Java (n'oubliez pas d'ajouter import java.util.Scanner;) tout au début de votre code.

Question 11: (5 minutes) **Format d'impression (Python uniquement)**

Créez et assignez des valeurs à 2 variables prenom (str) et age (int), puis affichez : "Je m'appelle prenom et j'ai age ans". Gérez le format de l'impression via l'opérateur +, puis en utilisant la fonction format().



Conseil

N'hésitez pas à consulter ce lien pour plus de détails concernant l'utilisation de la fonction format(): https://docs.python.org/fr/3.5/library/stdtypes.html#str. format

Question 12: (*3 minutes*) **Type** (**Python uniquement**)

Déclarez deux variables nom (String) et age (int), puis affichez le type de chacune de ces deux variables.



Conseil

Vous pouvez contrôler le type de vos variables via la fonction type().

Question 13: (5 minutes) Conversion des variables (Type casting) (Java ou Python)

Il est possible de convertir une variable d'un certain type vers un autre type. Il est par exemple possible de changer un int en float ou un float en int. Cette opération se nomme le Type Casting.

Déclarez une variable nombre_entier de type int, puis une autre variable nombre_decimal de type float. Affichez nombre_entier en le convertissant en float et nombre_decimal en le convertissant en int.



Conseil

Utilisez la fonction int(float) et float(int) en Python / Utilisez (int) float et (float) int en Java.

Question 14: (*3 minutes*) **Calculs (multiplication) (Java ou Python)**

Créez 2 variables facteur_1 (= 11) et facteur_2 (= 3). Multipliez la première variable par la deuxième et stockez le résultat dans une nouvelle variable produit. Vous pouvez afficher les différentes variables pour voir leurs valeurs. Vous pouvez répéter l'exercice avec l'addition et la soustraction.

Conseil

L'opérateur de multiplication est le *, celui d'addition est le + et celui de soustraction est le -.

Question 15: (10 minutes) Calculs (division) (Java ou Python)

Créez 2 variables nb_bonbons avec pour valeur 11 et nb_personnes avec pour valeur 3. Divisez la première variable par la deuxième et stockez le résultat dans une nouvelle variable bonbons_personnes. Pour finir, calculez le nombre de bonbons restants via l'opérateur % (modulo) et stockez le résultat dans une nouvelle variable reste. Vous pouvez afficher les différentes variables pour voir leurs valeurs.

Conseil

Attention, en Python il existe 2 opérateurs de division, / effectue une division classique, tandis que // effectue une division entière. En Java, si vous travaillez uniquement avec des int, / effectuera une division entière tandis que si vous travaillez avec au moins un float, / effectuera une division classique. Vous pouvez aussi formater le type du résultat lorsque vous créez une variable.

Question 16: (5 minutes) Calculs (incrémentation / décrémentation) (Java ou Python) Gardez vos variables de l'exercice précédent (Calculs (division) (Java ou Python)), augmentez la valeur de *nb_bonbons* de 1, et diminuez celle de *nb_personnes* de 1.

Conseil

Vous pouvez utiliser les opérateurs += et -= en Python, et les opérateurs ++ et - - en Java.

Question 17: (5 minutes) Manipulation des chaînes de caractères (indexation) (Java ou Python) Créez une variable mon_mot de type chaîne de caractères avec pour valeur "Hard But Cool!!". Créez ensuite une variable *premiere* contenant la première lettre de *mon_mot* en utilisant l'indexation. Créez enfin une variable derniere contenant la dernière lettre de mon_mot en utilisant l'indexation. Affichez les résultats. Qu'obtenez-vous?



Conseil

Pour Python, utilisez [], et pour Java, utilisez la fonction substring() ainsi que la fonction length() qui permet d'obtenir la taille d'un élément.

Question 18: (5 minutes) Manipulation des chaînes de caractères (indexation 2) (Java ou Python) Gardez votre variable, mon_mot et créez une variable lettre_5 contenant la cinquième lettre de mon_mot en utilisant l'indexation. Créez ensuite une variable lettre_9_13 contenant les lettres 9, 10, 11, 12, 13 de mon_mot. Afficher les résultats et voyez ce que vous obtenez.

Conseil

Attention, ici les espaces comptent comme des lettres!

Pour Python, utilisez [:], et pour Java, utilisez la fonction substring().

Question 19: (5 minutes) Les fonctions (fonctions basiques) (Java ou Python))

Définissez une fonction nommée ping() qui, lorsqu'elle est appelée, affiche "pong". Appelez la plusieurs fois et observez le résultat.

Conseil

- Référez vous aux diapositives du cours pour la création et l'appel des fonctions.
- Vous pourriez utiliser une boucle for pour effectuer plusieurs appels à la fonction ping().

Question 20: (5 minutes) Les Fonctions (Fonctions Aire et Périmètre) (Java ou Python)

Définissez deux fonctions nommées aire() et perimètre() qui prennent un argument (rayon) et renvoient respectivement l'aire et le périmètre d'un cercle. Stockez les résultats dans des variables aire et perimetre et affichez le contenu de ces variables.

- Référez vous au cours pour la création et l'appel des fonctions.
- Pour retourner une valeur au lieu de l'imprimer, utilisez le mot clé return (pour Python et
- Pour rappel, le périmètre d'un cercle s'obtient en utilisant la formule $P=2*\pi*r$ et l'aire s'obtient en utilisant la formule $A = r^2 * \pi$.

4 Opérateurs et conditions Booléennes (Python uniquement)

Le principe d'une valeur booléenne est qu'elle ne puisse contenir que 2 valeurs possibles, soit True, soit False. Il est possible de les définir en leur associant une de ces valeurs d'emblée ou de les obtenir en effectuant une comparaison. Pour ce faire, il faut utiliser des opérateurs booléens. Voici les plus utilisés : == (est égal), ! = (n'est pas égal), < (est strictement plus petit), <= (est plus petit ou égal), > (est strictement plus grand), >= (est plus grand ou égal). Si la condition est satisfaite, on obtiendra True, si elle ne l'est pas, on obtiendra False. L'utilisation de l'opérateur not inversera le résultat.

Dans les exercices suivants, vous devrez anticiper la valeur que la console va vous donner (résultat du(des) print(s)).



Utilisez les tables de vérité présentées à la diapositive 23 du cours.

Question 21: (5 minutes) Qu'affichera le programme suivant?

```
1
    a = 3
    \mathbf{b} = 2
2
    c = 6
3
4
    d = c >= b
5
    print(a==b)
    print(a*b==c)
7
    print(d)
8
    print(a \le b)
    print(not d)
```

Question 22: (5 minutes) Qu'affichera le programme suivant?

```
    a = 3
    b = 2
    c = 6
    d = c >= b
    print(a==b or d)
    print(a==b and d)
    print(a==b or a==c or False or d)
    print(a < c and not (b == 2 and not d))</li>
```

Question 23: (*5 minutes*) Qu'affichera le programme suivant?

```
a = True
     b = False
2
 3
     c = True
     if a or (b and not c):
5
       print("output 1")
6
     if b!=c:
       print("output 2")
7
    if a or (not b != (c \text{ and } a)):
9
       print("output 3")
10
   if not a or (b==c and not b):
       print("output 4")
```

5 Conditions

Le but de cette section est de vous entraîner à la lecture de code, la compréhension des opérateurs booléens et au "case switching" à travers le branchement conditionnel.

Question 24: (10 minutes) Branchement conditionnel en Java Qu'affiche le programme suivant?

```
int numero_mois = 7;
 2
 3
      switch(numero_mois) {
 4
 5
        case 1:
 6
              System.out.println("Janvier");
 7
              break;
 8
        case 2:
 9
              System.out.println("Février");
10
11
        case 3:
              System.out.println("Mars");
12
13
              break;
14
        case 4:
15
              System.out.println("Avril");
16
              break;
        case 5:
17
18
              System.out.println("Mai");
19
              break:
20
        case 6:
21
              System.out.println("Juin");
22
              break:
23
        case 7:
24
              System.out.println("Juillet");
25
              numero_mois = 9:
26
27
              System.out.println("Aout");
28
        case 9:
29
              if (numero\_mois == 8){
30
               System.out.println("Septembre");
31
               break;
32
               else {
33
34
               System.out.println("Décembre");
35
               numero_mois = 13;
36
               break:
37
38
        case 10:
39
              System.out.println("Octobre");
40
              break;
41
        case 11:
              System.out.println("Novembre");
42
43
              break;
44
        case 12:
45
              System.out.println("Décembre");
46
              break:
47
        default:
48
              System.out.println("Ce n'est pas un mois.");
     }
49
```

- break indique que l'on sort de l'accolade. Les cas suivants ne seront pas traités.
- L'absence de break indique que l'on va rentrer dans tous les cas suivants, jusqu'à enfin atteindre un break.
- Lorsque l'on pose case n où n est un nombre cela est équivalent au test n == numero_mois. Ce test est aussi valable si on cherche à comparer des chaînes de caractères (par exemple si numero_mois = "Juin", à ce moment là n sera aussi une chaîne de caractères).

Question 25: (5 minutes) Conditions en Python

Qu'affiche le programme suivant?

```
teacher = "Garbinato"
     assistant = "Diallo"
 3
     lesson = "ACT"
     if ((teacher == "Diallo") or not(teacher == "Garbinato")) and (lesson == "ACT"):
 4
 5
       print("Alpha Diallo et Benoît Garbinato")
 6
     elif ((teacher == "Bonvin") and (lesson == "ACT")) or (teacher == assistant):
       print("Arnaud Bonvin")
     elif (teacher == "Etienne Bruno" or assistant == "Diallo") and (lesson == "ACT" and teacher == "Olivier"):
 8
 9
       print("Etienne Bruno")
     elif ((teacher == "Garbinato" or lesson == "INF") and assistant == "Diallo") or (lesson == "ACT" and teacher == "Diallo"):
10
       print(teacher + ": Professeur du cours Algorithmique et pensée computationelle.")
11
12
     else:
13
       print("Benoît")
```

•

Conseil

- Il faut vérifier la condition de chaque cas de façon linéaire.
- Une fois une condition vérifiée, toutes celles d'après ne sont pas traitées.

Question 26: (5 minutes) Conditions IF (Python et Java)

Créez un petit script pour aider un videur à décider s'il laisse entrer une personne ou non dans une boite de nuit, et s'il doit lui donner un bon pour une boisson gratuite. Dans cette boite de nuit, les personnes de moins de 21 ans ne peuvent pas entrer, et les personnes ayant un âge compris entre 21 et 25 ans reçoivent un bon pour une boisson gratuite à l'entrée.

Demandez à l'utilisateur d'entrer l'âge de la personne, stockez cette valeur dans une variable nommée *age*. Si la personne a moins de 21 ans, affichez : "Entrée : Pas OK". Si la personne a entre 21 et 25 ans, affichez : "Entrée : OK, Boisson : OK". Si la personne a plus de 25 ans, affichez : "Entrée : OK, Boisson : Pas OK".

0

- Commencez par utiliser la fonction input() pour Python / la classe Scanner pour Java (n'oubliez pas le import java.util.Scanner;)
- Au niveau des conditions, commencez par traiter le cas dans lequel la personne a moins de 21 ans, puis traitez le cas dans lequel la personne a plus ou moins de 25 ans.