



# 5IPRO

## Principes algorithmiques et programmation

Benjamin Delbar



# Cours 1

Qui suis-je?

Objectif du cours

Evaluations

Formulaire

Principes algorithmiques



# Qui suis-je?

Full Stack Dev

Chargé de cours en 2ème Bac info

Coordinateur & chargé de cours section web dev

Remplacement de Rudy Renard

[benjamin.delbar@ifosup.wavre.be](mailto:benjamin.delbar@ifosup.wavre.be)



## Objectif du cours - programme

- ◆ d'identifier les différents langages de programmation existants ;
- ◆ de choisir le langage de programmation approprié à son application ;
- ◆ de mettre en œuvre une méthodologie de résolution de problème (observation, résolution, expérimentation, validation) et de la justifier en fonction de l'objectif poursuivi ;
- ◆ de concevoir, construire et représenter des algorithmes, en utilisant :
  - ◆ les types de données élémentaires,
  - ◆ les figures algorithmiques de base (séquence, alternative et répétitive),
  - ◆ les instructions,
  - ◆ les portées des variables,
  - ◆ les fonctions et procédures,
  - ◆ la récursivité,
  - ◆ les entrées/sorties,
  - ◆ les fichiers,
  - ◆ les structures de données de base (tableaux et enregistrements) ;
- ◆ de traduire de manière adéquate des algorithmes en respectant les spécificités du langage utilisé ;
- ◆ de documenter de manière complète et précise les programmes développés ;
- ◆ de produire des tests pour valider les programmes développés.



## Objectifs du cours - capacités terminales

- ♦ de mettre en œuvre une stratégie cohérente de résolution du problème posé ;
- ♦ de concevoir, de construire et de représenter l' (les) algorithme(s) correspondant(s) ;
- ♦ de justifier la démarche algorithmique et les choix mis en œuvre ;
- ♦ de développer des programmes en respectant les spécificités du langage choisi ;
- ♦ de mettre en œuvre des procédures de test.

Pour la détermination du degré de maîtrise, il sera tenu compte :

- ♦ de la qualité et de la pertinence de la démarche algorithmique,
- ♦ de la rigueur et du respect des spécificités du langage,
- ♦ du degré de précision et de la clarté dans l'emploi des termes techniques.



# Evaluations

Évaluations tout au long de l'année

Évaluations de rattrapage

Examen final\*

Répartition Évaluations / Examen\*

*\*Dans le cas ou Mr Renard ne reviendrait pas avant la fin du module*



# Formulaire

Pour mieux vous connaître

Vos attentes

Votre situation



# Principe algorithmique

1. Algorithme? Qu'est ce que c'est?
2. Programmes informatique
3. Langage de programmation
4. Récapitulons
5. Exemple & Exercices
6. Les conditions
7. Flowcharts
8. Opérateurs relationnels
9. Opérateurs arithmétiques





# Algorithme? Qu'est-ce que c'est?

Un **algorithme** est composé d'instructions et d'opérations réalisées, dans un ordre précis, afin de produire un résultat, et souvent résoudre un problème plus ou moins complexe

Ex: Recette de cuisine, Mode d'emploi pour monter un meuble, ...

Au niveau informatique : Un **algorithme** est une suite finie et non ambiguë d'instructions et d'opérations permettant de résoudre une classe de problèmes



# Programmes informatique

Forme nécessaire des algorithmes pour être exécuté par un ordinateur

Suite d'instruction qui explique à l'ordinateur les opérations à exécuter



# Langages de programmation

Langage utilisé pour réaliser un programme

Les algorithmes sont **indépendants** du langage de programmation utilisé

Un même algorithme peut être réalisé dans différents langages

Ex: C, C++, C#, Java, Javascript, PHP, Python, ...



# Récap

Algorithme = ?

Programmes informatique = ?

Langages de programmation = ?

## Exemple

Faire un café

- Déterminer les étapes
- Suivre cela correctement
- Apprendre des erreurs

=> **Fondements de la  
conception d'algorithmes**





# Exemple

Itération & raffinement

1. Chauffer de l'eau
2. Ajouter le café dans la tasse
3. Ajouter l'eau dans la tasse



# Exemple

1. Chauffer de l'eau
  - 1.1 Remplir la bouilloire d'eau
  - 1.2 Allumer la bouilloire
  - 1.3 Attendre que la bouilloire s'éteigne et que l'eau soit chaude



## Exemple

### 2. Ajouter le café dans la tasse

2.1 Ouvrir le récipient avec le café

2.2 Prendre du café dans le récipient avec une cuillère

2.3 Mettre la cuillère de café dans la tasse

2.4 Refermer le récipient avec le café





# Exemple

3. Ajouter l'eau dans la tasse

3.1 Verser l'eau de la bouilloire dans la tasse



On pourrait encore raffiner nos instructions

1. Chauffer de l'eau

1.1 Remplir la bouilloire d'eau

1.2 Allumer la bouilloire

1.3 Attendre que la bouilloire s'éteigne et que l'eau soit chaude

2. Ajouter le café dans la tasse

2.1 Ouvrir le récipient avec le café

2.2 Prendre du café dans le récipient avec une cuillère

2.3 Mettre la cuillère de café dans la tasse

2.4 Refermer le récipient avec le café

3. Ajouter l'eau dans la tasse

3.1 Verser l'eau de la bouilloire dans la tasse



# Exercice

Rédiger, en groupe, les instructions nécessaire pour que je sorte de la classe



## Exercices

Rédiger les instructions permettant d'obtenir le résultat par rapport aux entrées

1

input a	input b	output
5	2	10
9	7	63

2

input a	input b	output
15	3	3
12	23	12

3

input a	input b	output
15	3	15
12	23	23

4

input a	input b	output
5	2	25
6	3	216

5

input a	input b	output
"John"	"Doe"	"Bonjour John Doe"
"Marie"	"Dupont"	"Bonjour Marie Dupont"



# Les conditions

Algorithme peu flexible

- Instructions exécutées une seule fois
- Dans l'ordre
- La dernière instruction => fin de l'algorithme

Exemple : que faire si le récipient à café est vide?



# Les conditions

Si .... Alors ... (*if ... then ...*)

**Si** récipient à café est vide

**Alors** aller chercher un autre récipient plein



## Les conditions

Si ... Alors .... Sinon ... (*if ... then ... else ...*)

**Si** récipient à café est vide

**Alors** aller chercher un autre récipient plein

**Sinon** utiliser le récipient actuel



## Les conditions

Si ... Alors .... Sinon si ... Alors ... Sinon (*if ... then ... else if ... then ... else ...*)

**Si** récipient à café est vide

**Alors** aller chercher un autre récipient plein

**Sinon si** récipient à café est presque vide

**Alors** utiliser le récipient actuel et aller chercher un autre récipient plein

**Sinon** utiliser le récipient actuel





# Les conditions : exercices

Réalisez les instructions nécessaire

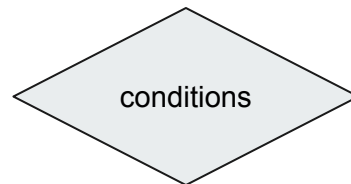
input a	input b	output
15	3	3
12	23	12


input a	input b	output
15	3	15
12	23	23



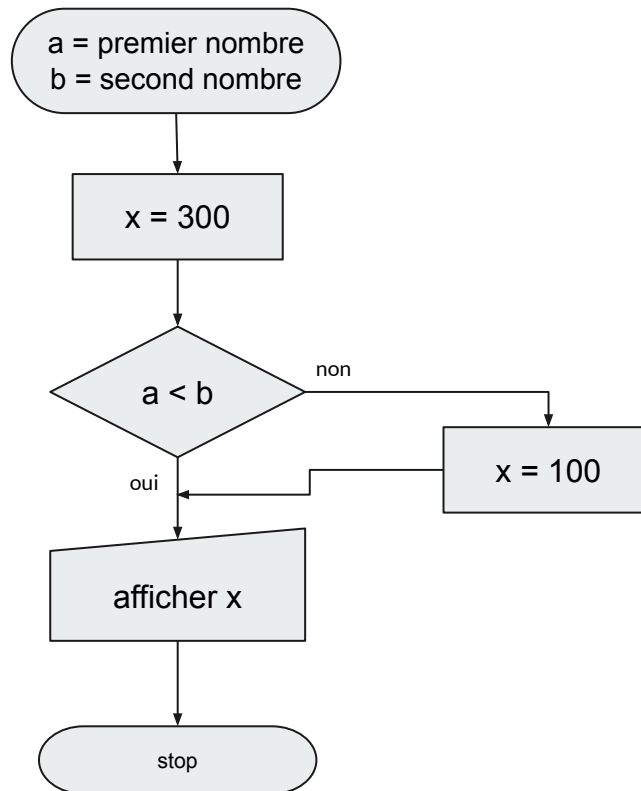
# Flowcharts

Permettre de symboliser nos instructions





a	b	x
5	10	
10	5	
7	7	





Réalisez le flowchart expliquant l'algorithme fournissant les résultats suivant

a (entrée)	b (entrée)	x (sortie)
5	2	10
9	7	63

a	b	c	x
5	10	15	
15	5	10	
15	10	5	

