

---

# INF2010

# Algorithmes

# et Structures de données

## Introduction

---

# Considérations officielles

---

- INF2010 – Algorithmes et Structures de données  
→ Cours de deuxième année
  - Réussir le cours est requis pour suivre de nombreux cours du baccalauréat en génie informatique et génie logiciel, (dont certains à option):
    - LOG3210 (Éléments de langage et compilateurs)
    - INF3710 (Fichiers et bases de données)
    - INF4705 (Analyse et conception d'algorithmes)
    - INF8701 (Simulation des systèmes à événements discrets)
-

# Considérations officielles (...)

---

- Évaluation:
    - Laboratoires 30% (5% par laboratoire)
    - Contrôle périodique 30%
    - Examen final 40%
  - Laboratoires:
    - 6 séances (une fois aux deux semaines)
    - La présence aux laboratoires est OBLIGATOIRE!!!
    - Les laboratoires sont réalisés en équipe de deux
    - Le logiciel Eclipse est utilisé au laboratoire
-

# Considérations officielles (...)

---

- Coordonateur  
Ettore Merlo
- Coordonnées:  
Bureau M-4105  
Extension téléphonique 5758  
E-mail : **ettore.merlo@polymtl.ca**

# Considérations officielles (...)

---

- Chargé de cours:  
Tarek Ould Bachir  
→ Ingénieur R&D OPAL-RT Technologies
- Coordonnées:  
E-mail : **tarkoo@gmail.com**
- Disponibilités  
Après la séance de cours ou sur RDV (par courriel)

# Considérations officielles (...)

---

- Site web du cours:  
[www.moodle.polymtl.ca](http://www.moodle.polymtl.ca)
  - Documentation:
    - Le livre de référence (en vente à la Coop)
    - Acétates du cours (disponibles sur le site web du cours)
  - Plan de cours:  
Contient toute l'information présentée ici et plus.  
Disponible sur le site web du cours...
  - Coordonateur du cours:  
Le professeur Ettore Merlo
-

# Ce que nous allons étudier

---

- Les structures de données (SD) séquentielles
  - La manipulation des SD séquentielles
  - Algorithmes de tri
  - Arbres binaires/Arbres binaires de recherche
  - Arbres équilibrés (AVL, Splay,...)
  - File de priorité (monceaux)
  - Algorithmes sur les chaînes de caractères
  - Graphes et algorithmes sur les graphes
  - Ensembles disjoints
-

# Après le cours, vous pourrez

---

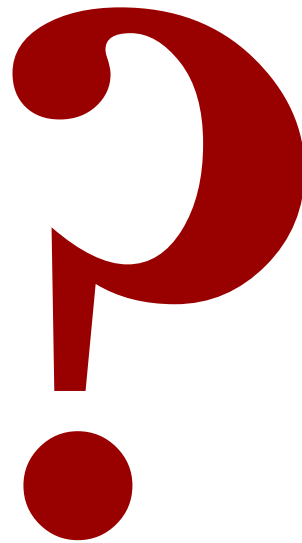
- Comprendre les fondements sur lesquels reposent les SD et les algorithmes les plus communément employés pour manipuler les manipuler
  - Savoir utiliser et adapter ces SD pour la résolution d'un problème spécifique
  - Être capable de concevoir de nouvelles SD pour la résolution d'un problème spécifique
-



# Commençons un peu...

---

Qu'est ce qu'une structure de donnée?



# La structure de données

---

- Est-ce ça?

```
// Déclare Etudiant
struct Etudiant {
    int age;
    float moyenne;
    char nom[25];
};
```

# La structure de données

---

- Définition

Structure logique servant à contenir des données et dont le but est d'en simplifier le traitement.

On entend par traitement de données, l'accès en lecture et écriture (get/set), l'insertion ou le retrait (add/remove), le tri, etc.

# Algorithmes

---

Comment mesurer la performance d'un algorithme?

- Temps d'exécution ?
  - Dépend de la machine
  - Dépend du langage
  - Requiert des tests empiriques
- Nombre de lignes de code ?
  - On a mesuré par le passé la complexité d'un programme par le nombre de lignes de codes. Est-ce vraiment pertinent encore aujourd'hui?

# Algorithmes

---

## Notation asymptotique

- $O( g(n) )$ 
  - $O( n )$
  - $O( n^2 )$
  - $O( \log(n) )$
  - $O( n \log(n) )$
- Qu'est-ce donc  $n$  ?

# Algorithmes

---

Quelle est la complexité asymptotique du bout de code suivant:

```
for (int i=0; i<n; i++)  
    for (int j=0; j<n; j++)  
        FaisQuelqueChose();
```

FaisQuelqueChose() a un temps d'exécution unitaire

# Algorithmes

---

Quelle est la complexité asymptotique du bout de code suivant:

```
for (int i=0; i<n; i++)  
    for (int j=i; j<n; j++)  
        FaisQuelqueChose();
```

FaisQuelqueChose() a un temps d'exécution unitaire

# Algorithmes

---

Quelle est la complexité asymptotique du bout de code suivant:

```
for (int i=0; i<n; i++)  
{  
    for (int j=i; j<n; j++)  
        FaisQuelqueChose();  
  
    for (int k=0; k<n; k++)  
        FaisAutreChose();  
}
```

FaisQuelqueChose() et FaisAutreChose() ont un temps d'exécution unitaire



# Algorithmes

---

Quelle est la complexité asymptotique du bout de code suivant:

```
int calcul( int n )
{
    if( n = 1 )
        return 1;
    else
        return calcul( n/2 ) + 1;
}
```

# Langage JAVA

---

Pour enseigner les SD, on préférera un langage de haut niveau comme C++ ou JAVA.

Connaissez-vous JAVA?



# Spécificités du JAVA

---

Qu'est-ce que c'est ?

- Le JAVA est un langage de programmation orienté objet
  - Le JAVA reprend la syntaxe du C++
  - Le JAVA exploite une machine virtuelle (JVM) pour être multi plate-forme
-

# Spécificités du JAVA

---

## Spécificités:

- En JAVA, tout est objet, à l'exception des quelques types primitifs
  - En JAVA, tout est référence (et il n'y a pas de pointeur à proprement parler)
  - Le JAVA exploite un ramasse-miette qui facilite grandement la vie du développeur
-

# Spécificités du JAVA

---

## Exemple de code:

```
public static int OuSontLesReferences( int n )
{
    Integer N1 = new Integer( n );
    Integer N2 = new Integer( 2*n );

    N1 = N2;
    N2 = N1;
    N2 = N1 + N2;

    return N2.intValue();
}
```

---

# Spécificités du JAVA

---

## Spécificités:

- En JAVA, l'héritage multiple n'existe pas:
  - Un objet hérite d'un seul parent
  - Un objet peut cependant implémenter autant d'interfaces que désiré
  - Un objet implémentant une interface doit implémenter toutes les méthodes de l'interface

# Spécificités du JAVA

---

Exemple de code:

```
public class Vecteur3D{
    protected int a, b, c;
    Vecteur3D( int a, int b, int c) {
        this.a = a;
        this.b = b;
        this.c = c;
    }

    public String toString(){
        return "(" + a + ", " + b + ", " + c + ")";
    }
}
```

---

# Spécificités du JAVA

---

Exemple de code:

```
public interface Arith
{
    public int  somme( );
    public int  produit( );
}
```



# Spécificités du JAVA

---

Exemple de code:

```
public class Vecteur3DArith extends Vecteur3D implements Arith{

    public Vecteur3DArith(int a, int b, int c) {super(a, b, c);}

    public int somme(){return a + b + c;}

    public int produit(){return a * b * c;}

    public String toString(){
        return super.toString() +
            "\n Somme: " + somme() +
            "\n Produit: " + produit();
    }
}
```

---

# Spécificités du JAVA

---

## Spécificités:

- Le JAVA dispose de nombreuses librairies très riches et puissantes. Il vous revient de les découvrir pour en tirer le meilleur profit:
  - Fouillez l'Internet
  - Prêtez attention aux classes vues en classe
  - Servez-vous de l'aide contextuelle de l'IDE (notamment l'auto-complétion)