

ITI 1521. Introduction à l'informatique II

Exigences du cours

by

Marcel Turcotte

Version du 6 janvier 2020

Préambule

Préambule

Objectifs d'apprentissage

Objectifs d'apprentissage

- ❖ **Informer** sur les exigences du cours.
- ❖ **Expliquer** les objectifs d'apprentissage généraux dans ce cours.
- ❖ **Connaître** les règlements universitaires sur la fraude scolaire.

Lectures :

- ❖ Ce document, ainsi que ceux-ci :
 - ❖ Plan de cours
 - ❖ Intégrité dans les études

Préambule

Plan du module

Plan

- 1 Préambule
- 2 Plan de cours
- 3 Intégrité dans les études
- 4 Contenu
- 5 Épilogue
- 6 Prologue

Marcel Turcotte : en bref

1965- Né à **Montréal**

1995 Ph. D. en informatique, Université de **Montréal**

1995–97 Université de la **Floride** (É.-U.), Département de Chimie

1997–00 Imperial Cancer Research Fund, **Londres**, Angleterre

2000- Université d'**Ottawa**

2006–08 **Coordonnateur** du programme **coop** pour l'informatique

2010–11 **Coordonnateur** du programme de maîtrise en
bio-informatique

2012–18 **Vice-doyen** aux études de premier cycle



Source : blog.f1000.com

Marcel Turcotte : en bref

1. ARN et ADN, régulation de l'expression des gènes, structure secondaire, inférence de motifs, algorithmique du texte
2. **Logiciels** : MC-Sym (co-auteur), eXtended-Dynalign, Profile-Dynalign, Seed, ACSEA, ModuleInducer, RiboFSM, MotifGP, Seed_w
3. Conception d'algorithmes et de structures de données, intelligence artificielle appliquée et apprentissage machine



**Rocket science is for kids
Bioinformatics is for scientists**



Source : QIAGEN/CLC bio

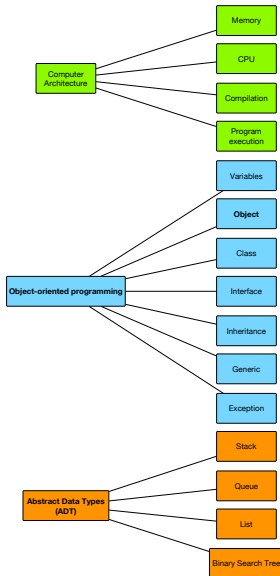
Plan de cours

Computer
Architecture

Object-oriented programming

**Abstract Data Types
(ADT)**

Contenu du cours





Prenez note !

- ❖ « Le professeur doit fournir un **plan de cours** lors de la première rencontre avec les étudiants au début de chaque cours. » (Règlement scolaire 8.5)
- ❖ Ce plan de cours **demeurera inchangé** pour tout le reste du semestre !

Évaluations

- ✦ Examen **mi-session** : 25 %
- ✦ Examen **final** : 40 %
- ✦ **Devoirs** (4) : 25 %
- ✦ **Laboratoires** (10–12) : 10 %

Vous devez obtenir **au moins 50% aux examens** pour passer le cours, un échec aux examens entraînera un échec pour le cours !

Examens

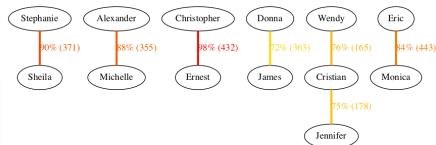
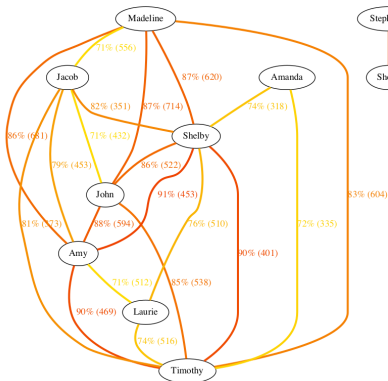
- Mi-session** ❑ Dimanche **1 mars 2020, 10 h à 12 h**
- ❑ Livres fermés
- Final** ❑ Livres fermés

Intégrité dans les études



Ceci s'applique à tout le monde

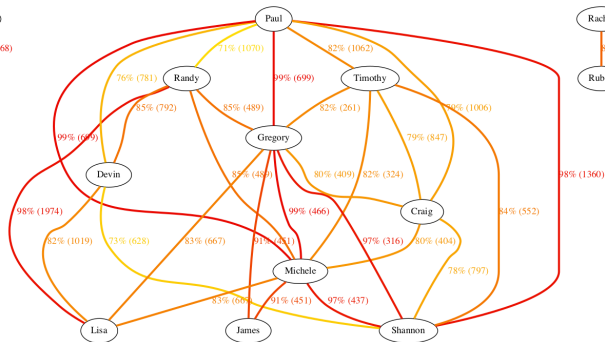
- L'an passé, plus de **40 étudiants** ont eu une allégation de fraude contre eux pour les devoirs 1 ...



Ceci s'applique à tout le monde

■ ... et 2 !

- Chaque lien montre la relation entre une **paire de soumissions**.
 - Les étiquettes indiquent le **degré de similitude** et le **nombre de lignes de code** (entre parenthèses).



Intégrité dans les études

Fraude scolaire : « (...) tout acte commis par un étudiant qui peut avoir pour **résultat la falsification de son évaluation scolaire** ou de celle d'un autre étudiant.

Voici des **exemples** :

- ❖ remettre un travail fait **par quelqu'un d'autre** ou faire le travail **pour quelqu'un d'autre** ;
- ❖ **falsifier** une évaluation scolaire ;
- ❖ présenter un travail **acheté sur Internet** ;
- ❖ **plagier les idées**.

Règlements :

- ❖ <http://www.uottawa.ca/administration-et-gouvernance/reglement-scolaire-14-autres-informations-importantes>

Bonnes pratiques

- ❖ Ne publiez jamais votre code source sur **GitHub** publiquement avant la date limite pour la soumission du devoir.
- ❖ Vous pouvez publier votre solution **après la date limite** de soumission.
 - ❖ Cependant, vous devriez retirer votre **numéro d'étudiant** de l'Université d'Ottawa, car quelqu'un d'autre pourrait utiliser cette information à mauvais escient.
 - ❖ Vous devez aussi être prudent, dans certains cas vous publiez du code source qui **n'est pas le vôtre**.
 - ❖ En particulier, vous pourriez publier le code source qui a été **développé par les instructeurs**.

Vous êtes la victime

- ❖ La première **victime** en copiant la solution de quelqu'un d'autre, **c'est vous**.
- ❖ Vous n'apprendrez pas «**comment apprendre**».
- ❖ **Peut-être** comprenez-vous la solution que vous avez copiée.
 - ❖ Cependant, vous n'aurez pas réussi à apprendre **comment créer ces solutions en premier lieu**.
 - ❖ Bien sûr, cela aura un impact sur votre capacité à passer les **examens**.
 - ❖ Mais cela aura aussi un impact sur votre capacité à trouver l'**emploi de vos rêves**.
 - ❖ Les **entrevues** pour les postes en génie logiciel et en informatique sont très approfondies.
 - ❖ On vous demandera de **résoudre des problèmes** qui pourraient en fait être similaires à ceux que l'on trouve dans nos devoirs.

Soyez extrêmement prudent, dans certains cas, **les étudiants ne savaient pas que leur solution avait été copiée**, ce qui rend ce message important pour tous.

Fraude scolaire : scénarios

Levez la main, si vous avez l'intention de tricher dans ce cours !

- ❖ Quelqu'un vous demande l'accès à votre devoir
 - ❖ c'est un ami
 - ❖ il est trop tard, il ou elle ne terminera pas dans les temps. . .
 - ❖ **tous les intervenants obtiennent la note 0**
- ❖ Vous êtes bloqué
 - ❖ vous choisissez de collaborer avec une ou plusieurs équipes
 - ❖ **tous les intervenants obtiennent la note 0**
- ❖ **Autres sanctions**, les dossiers sont transmis au vice-doyen aux études de premier cycle : l'étudiant obtient **0**, pour **ce devoir**, pour **tous les devoirs**, pour **le cours**, suivre un **cours d'étiq**, **expulsion** de l'université, etc.

Analogie avec les sports de haut niveau



Politique sur l'utilisation des appareils électroniques



Opinion (que je partage) de Jacques Nantel des HEC

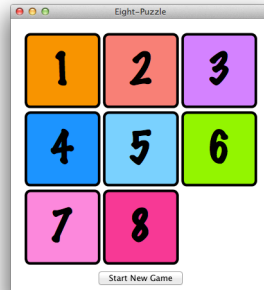
«[V]ous êtes devenus les fiduciaires des [dollars] confiés par les contribuables dans l'espoir que vous puissiez contribuer à améliorer notre société. Vous avez donc l'obligation de **travailler fort**, d'**apprendre** et de sortir plus instruits que vous ne l'étiez en y entrant.»

«La place que vous occupez», Opinion de Jacques Nantel des HEC publiée dans l'édition du 5 avril 2016 de **La Presse+**, Section débats, écran 6

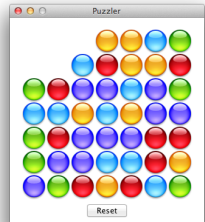
Contenu

Contenu

Devoirs



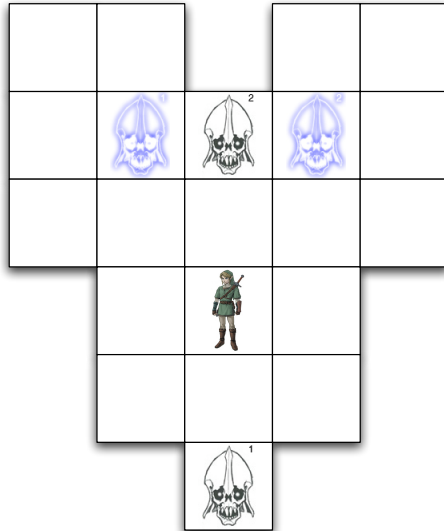
Devoirs



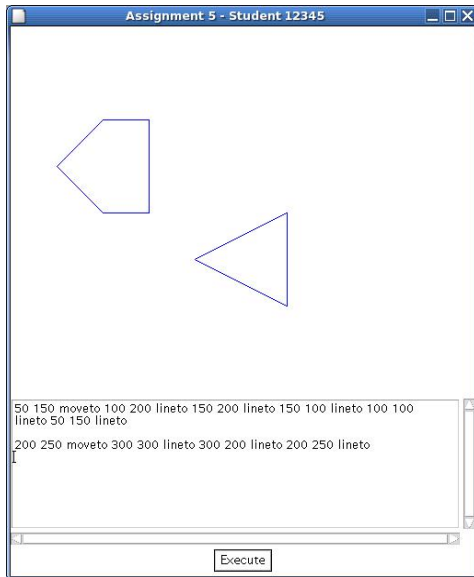
Devoirs

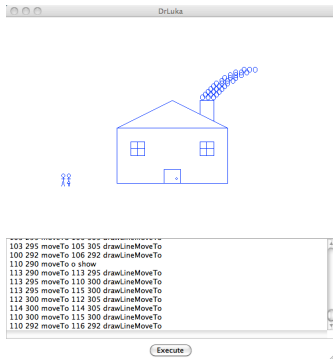


Devoirs

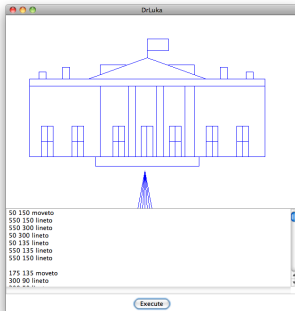


Devoirs

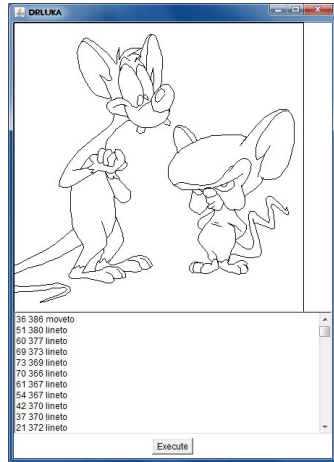




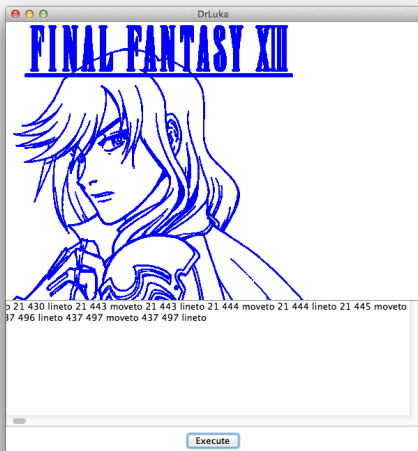
Samuel Bostock 2010



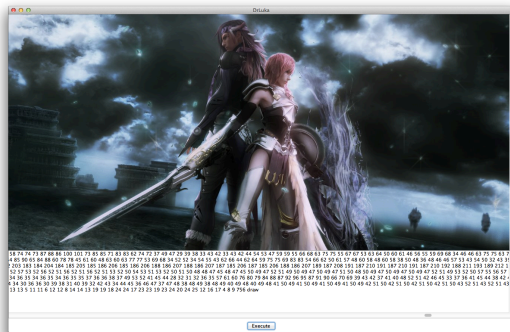
Liam Shea Williams 2010



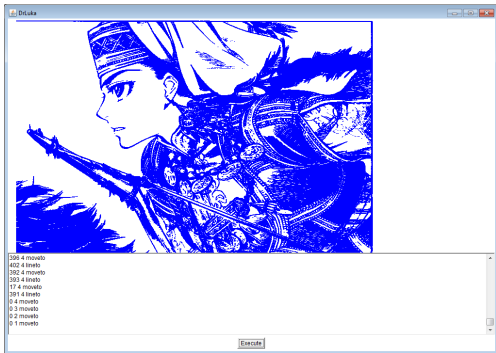
Quentin Smith 2011



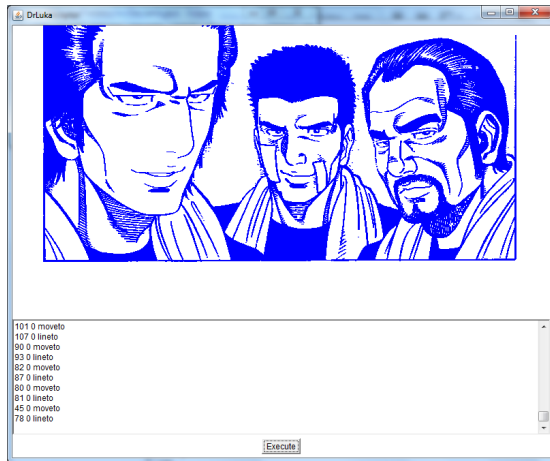
Olivier Gagnon 2012



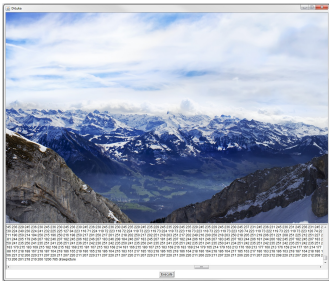
Olivier Gagnon 2012 (LVM Modifiée)



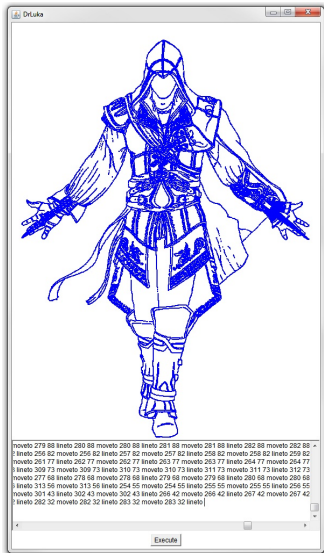
Liban Osman 2012



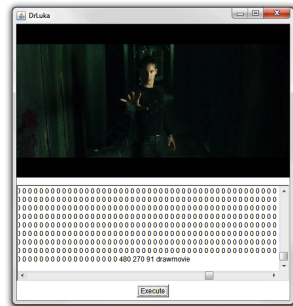
Liban Osman 2012



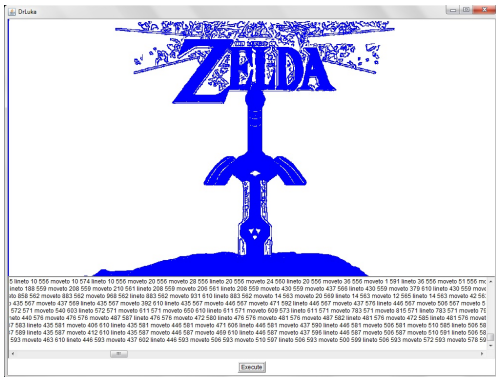
Matthew Horton 2012



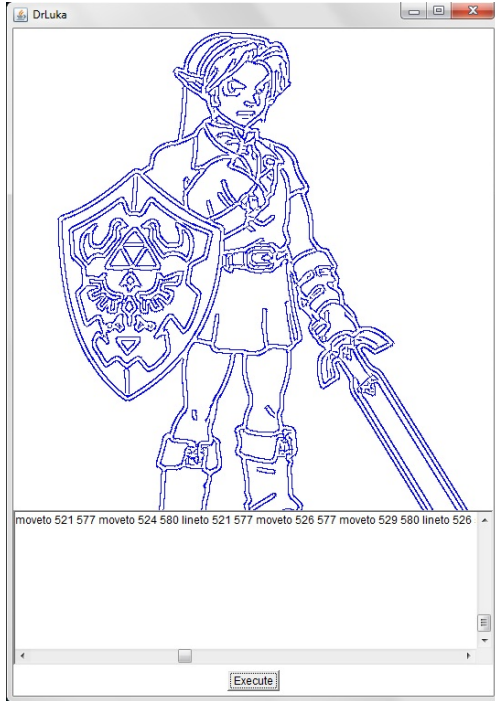
Matthew Horton 2012

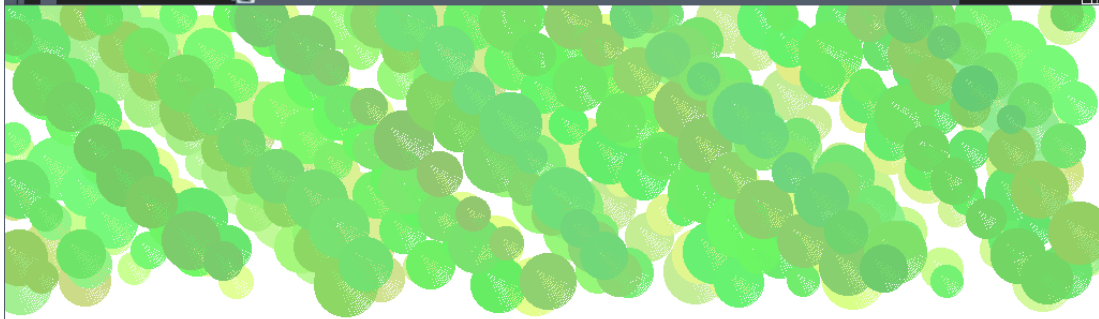


Matthew Horton 2012

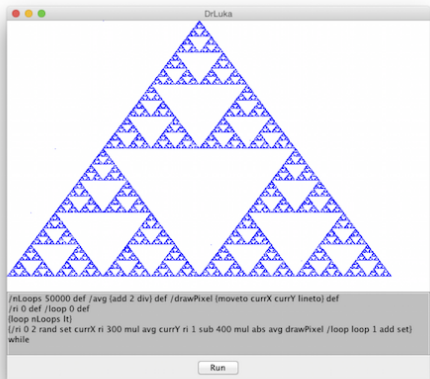


Jonathan Ermel 2012

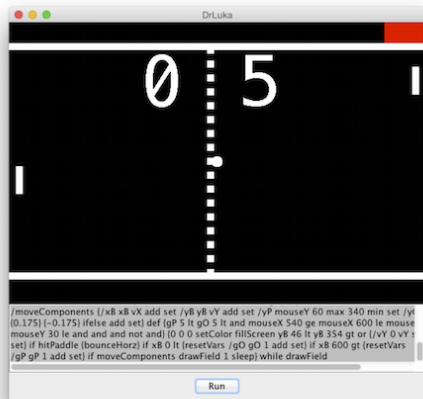




A	B	C	D	E	F	G	H	I	J
/C 0 cell if // 1 /B 2 popcell 30 rand 30 add /C 2 popcell /A 1 cell /C 0 cell 1 sub /C 0 popcell /A 0 cell		500	< # of circles						
/B /A 11 cell /B 3 cell /A 9 cell /B 1 popcell /C /A 11 cell /C 3 cell /A 9 cell /C 1 popcell /A 7 cell			< circle coords						
/C 2 cell if // /A 3 cell /C 2 cell 1 sub /C 2 popcell /A 2 cell			< circle size						
/B 1 cell /B 2 cell 4 div sub /C 1 cell /B 2 cell 4 div sub moveto /A 4 cell	1000	250	< canvas size	^rgb params					
/B 2 cell 0 360 arc /B 2 cell 1 add /B 2 popcell			/pop.luka dump						
100 155 rand add dup 100 155 rand add dup 3 1 roll 100 155 rand add dup 4 1 roll color /A 6 cell									
/E 2 popcell /E 1 popcell /E 0 popcell									
3 rand dup /E exch cell 20 rand 10 sub add dup 255 gt if // pop 255 // pop /A 8 cell /A 2 cell									
dup 100 lt if // pop 100 // pop exch /E exch popcell /E 0 cell /E 1 cell /E 2 cell color									
ge if // 0 // pop									
/A 5 cell /B 3 cell rand /B 1 popcell 0 /C 1 popcell /A 0 cell									
/B 2 cell dup 20 rand 20 add add exch 2 div sub exch 1 cell add dup									



Matthew Pollex 2015



Matthew Pollex 2015 (AI Pong)

Objectifs d'apprentissage du cours

- ❖ L'objectif principal du cours est de savoir implémenter et utiliser les **types abstraits de données**, et en particulier, les listes, les piles, les files et les arbres binaires de recherche.
- ❖ Afin de développer des solutions élégantes et robustes, des connaissances suffisantes de la **programmation orientée objet** sont nécessaires : encapsulation, héritage et polymorphisme.
- ❖ Nous traiterons aussi des entrées-sorties, interfaces usager, et JUnit.

Difficultés

- ▣ Les concepts de **haut niveau** sont simples
- ▣ La difficulté principale consiste à **traduire les idées de haut niveau en implémentation concrète**

Google en mode croissance à Montréal

La Presse, publié le 1er février 2011

(<http://bit.ly/zspirb>)

« (...) de très **fortes habiletés de programmation** (...) On ne veut **pas seulement des gens qui ont des idées**. On veut quelqu'un capable de s'asseoir et de coder »

« La directive qu'on a reçue est d'**engager autant de bons employés qu'on peut en recruter**. Il n'y a pas de quota sur la quantité, mais nous avons un seuil de qualité très élevé. »

« Les équipes de Toronto, **Ottawa** et Waterloo bénéficieront aussi d'un coup d'accélérateur »

We tend to look at the teacher/learner relationship the wrong way around : it's not that the teacher teaches ; **it's that the student learns.**

Pragmatic Thinking and Learning — Andy Hunt

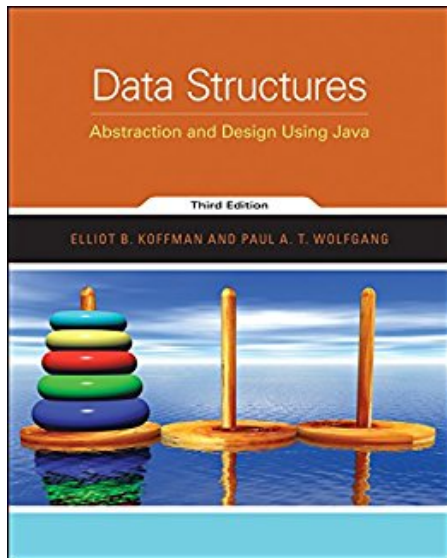
Difficultés

- ❖ Ne faites pas de copier-coller, **écrivez le code vous même !**
- ❖ **Faites tous les devoirs !**
- ❖ **Faites tous vos laboratoires !**
- ❖ Ayez une **attitude positive** face à la programmation !
- ❖ **Souriez !**

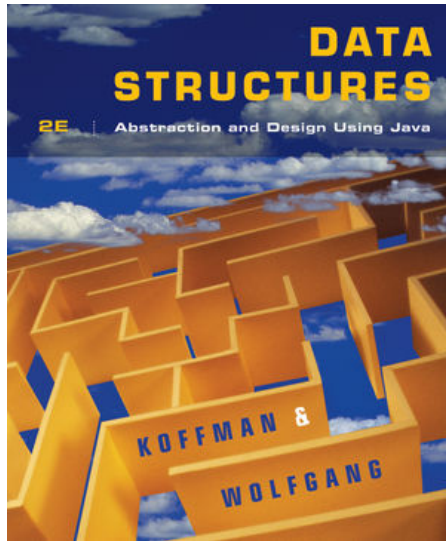
Difficultés (suite) : ITI1521 2016

20	19%	A+
4	4%	A
11	10%	A-
3	3%	B+
5	5%	B
6	6%	C+
3	3%	C
1	1%	D+
1	1%	D
6	6%	E
3	3%	F
4	4%	EIN
3	3%	ABS
38	35%	Abandon

- ✚ E. B. Koffman, P. A. T. Wolfgang. Data Structures : Abstraction and Design Using Java. John Wiley & Sons, 3e edition, 2016.



- ✚ E. B. Koffman and P. A. T. Wolfgang.
Data Structures : Abstraction and
Design Using Java. John Wiley & Sons,
2e edition, 2010.



Livre du cours (alternative)

Pourriez-vous recommander un volume plus abordable ?

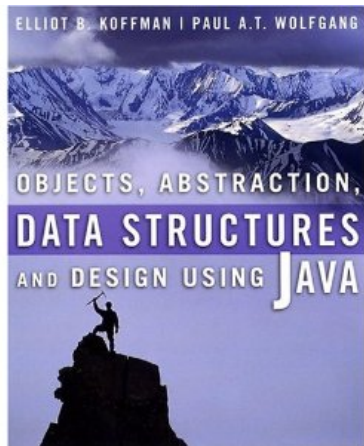
- ✚ E. Koffman and P. Wolfgang (2005)
Objects, Abstraction, Data Structures and Design : Using Java Version 5.0.
Wiley, 880 pages. (ISBN :
0-471-69264-6)



Livre du cours (alternative)

Pourriez-vous recommander un livre plus abordable ?

- ✚ E. Koffman and P. Wolfgang (2005)
Objects, Abstraction, Data Structures and Design : Using Java. Wiley, 864 pages. (ISBN : 0-471-46756-1)



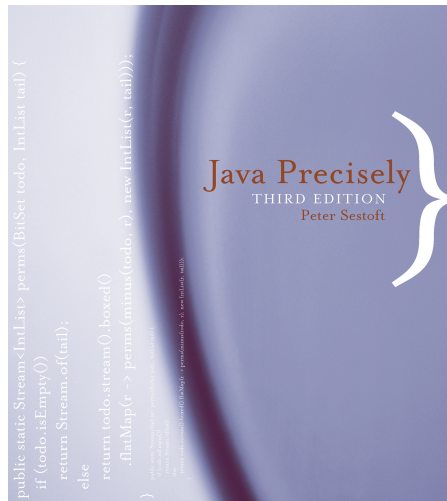
Livre du cours (alternative)

Pourriez-vous recommander un livre encore plus abordable ?

- ✚ Java Structures : Data Structures in Java for the Principled Programmer by Duane A. Bailey
 - ✚ www.cs.williams.edu/~bailey/JavaStructures/Book.html

Je ne connais pas le langage de programmation Java !

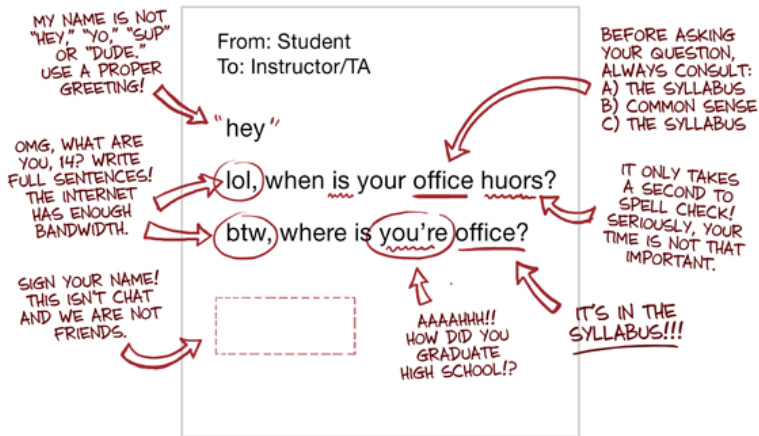
- P. Sestoft. Java Precisely. The MIT Press, 3e edition, August 2016. (~ 40 \$)



Épilogue

Netiquette

HOW TO WRITE AN E-MAIL TO YOUR INSTRUCTOR OR T.A.



JORGE CHAM © 2015

WWW.PHDCOMICS.COM

title: "How To Write An E-mail To Your Instructor Or Teaching Assistant" - originally published 4/22/2015

Prologue




Résumé

- ❖ Vous devez obtenir **au moins 50% pour la portion examens**
- ❖ **Faites** tous les devoirs
- ❖ **Faites** tous les laboratoires
- ❖ **Ne succombez** pas au plagiat

Prochain module

- ✚ Introduction à la **programmation orientée objet**

References I

-  E. B. Koffman and Wolfgang P. A. T.
Data Structures : Abstraction and Design Using Java.
John Wiley & Sons, 3e edition, 2016.
-  E. B. Koffman and Wolfgang P. A. T.
Data Structures : Abstraction and Design Using Java.
John Wiley & Sons, 2e edition, 2010.
-  P. Sestoft.
Java Precisely.
The MIT Press, 3e edition, 2016.



Marcel **Turcotte**

Marcel.Turcotte@uOttawa.ca

École de **science informatique** et de génie électrique (SIGE)
Université d'Ottawa