



POLYTECHNIQUE  
MONTRÉAL

## Questionnaire Contrôle Périodique 4

**LOG3430**

Sigle du cours

Identification de l'étudiant(e)		
Nom :	Prénom :	
Signature :	Matricule :	Groupe :

Sigle et titre du cours		Groupe	Trimestre
LOG3430 - Méthodes de test et de validation du logiciel		Tous	20151
Professeur		Local	Téléphone
Giuliano Antoniol		C-624	
Jour	Date	Durée	Heures
Vendredi	13 mars 2015	1 heure	
Documentation		Calculatrice	
<input type="checkbox"/> Aucune <input checked="" type="checkbox"/> Toute <input checked="" type="checkbox"/> Voir directives particulières		<input type="checkbox"/> Aucune <input checked="" type="checkbox"/> Toutes <input type="checkbox"/> Non programmable	Les connections aux réseaux sont interdits.
Directives particulières			
Toute documentation est permise, ainsi que les calculatrices, à l'exception toutefois des de tout dispositif connectés a Internet.			

<b>Important</b>	Cet examen contient <input type="text" value="1"/> exercice et <input type="text" value="0"/> question sur un total de <input type="text" value="5"/> pages (excluant cette page)
	La pondération de cet examen est de <input type="text" value="5"/> %
	Vous devez répondre sur : <input checked="" type="checkbox"/> le questionnaire <input type="checkbox"/> le cahier <input type="checkbox"/> les deux
	Vous devez remettre le questionnaire : <input checked="" type="checkbox"/> oui <input type="checkbox"/> non

L'étudiant doit honorer l'engagement pris lors de la signature du code de conduite.

## Exercice 1 – 20 points

Considérez les classes Address, Order, Person et Customer (voir le document annexe), il faut concevoir les tests unitaires.

1.1) (1 point) Selon le critère de Harrold McGregor, quelles sont les méthodes du Customer qui doivent être testées de nouveau ?

1.1.1 incrementRating

1.1.2 decRating

1.1.3 setRating

1.1.4 reset

1.1.5 \_\_\_\_\_

Justification We assume Person was tested first and now we test Customer ; rating is the only modified attribute of the father. Actually the two « setters » reset and setRating may not need to be retested. Yes there is a way to interact but both act as setters and nothing else of rating and trust respectively.

1.2 (1 point) Si on utilise le critère de Harrold McGregor, quelles sont les méthodes du Address qui doivent être testées de nouveau ?

1.2.1 \_\_\_\_\_

1.2.2 \_\_\_\_\_

1.2.3 \_\_\_\_\_

1.2.4 \_\_\_\_\_

Justification : No one no inheritance relation

1.3 (1 point) Si on veut tester les classes avec la méthode des tranches des données, spécifiez dans le tableau suivant combien de setter(s) et getter(s) il faut ajouter pour chaque classe :

Class	Setter	Getter
Address		1 (name)
Person	1(trust)	3(trust, rating name)
Customer	1(paidAmount)	3(customerNumber,order, paidAmount)
Order	3(customer, item, cost)	4 (customer, item, cost, totalToPay)

1.4 (2 points) Pour la classes Person, identifiez les tranches des données, ajoutez des getters si nécessaire

Attribute									
name	P()	P(..)	setName	getName					
address	P()	P(..)	setAddress	getAddress					
rating	P()	P(..)	setRating	incRating	decRating	getRating			
trust	P()	P(..)	incRating	Reset	setTrust	gestTrust	decRating		

NB : P() default constructor no param and P(...) constructor with parameters

1.5 (2 points) Pour la classe Customer, identifiez les « tranches des données », ajoutez des getters si nécessaire

Attribute									
customerNumber	C()	C(..)	setCustomerNumber	getCustomerNumber	incRating				
paidAmount	C()	C(..)	setPaidAmount	getPaidAmount					
order	C()	C(..)	placeOrder	getOrder					

NB : there are several problems with the constructors for example we need to fix the fact it does not initialize the order,

1.6 (2 points) Pour la classe Person, complétez le « MADUM », ajoutez des getters si nécessaire

(C : constructor, T : transformer, O : other, R : reporter/getter) :

	Person()	Person(..)	getAddress()	setAddress()	setName()	setRating()	incRating()	Reset()	decRating()
name	C	C			T				
address	C	C	R	T					
rating	C	C				T	T		T
trust	C	C					T	T	T

getter(s) à ajouter : getName, getRating and getTrust

1.7 (4 points) Pour la classe Person, donnez les séquences de test pour les « transformers » du rating et trust , utilisez les tableaux suivants : NB we must have 6 sequences for P() AND 6 for P(...) here the sequences for P()

rating :

P()	getRating	setRating	getRating	incRating	getRating	decRating	getRating		
P()	getRating	setRating	getRating	decRating	getRating	incRating	getRating		
P()	getRating	incRating	getRating	setRating	getRating	decRating	getRating		
P()	getRating	incRating	getRating	decRating	getRating	setRating	getRating		
P()	getRating	decRating	getRating	setRating	getRating	incRating	getRating		
P()	getRating	decRating	getRating	incRating	getRating	setRating	getRating		

trust : as for rating we must double the sequences one set for P(...)

P()	getTrust	incRating	getTrust	reset	getTrust	decRating	getTrust		
P()	getTrust	incRating	getTrust	decrating	getTrust	reset	getTrust		
P()	getTrust	reset	getTrust	incRating	getTrust	decRating	getTrust		
P()	getTrust	reset	getTrust	decRating	getTrust	incRating	getTrust		
P()	getTrust	decRating	getTrust	reset	getTrust	incRating	getTrust		
P()	getTrust	decRating	getTrust	incRating	getTrust	reset	getTrust		

1.8 (2 points) Pour la classe Customer, expliquez de quelle façon le « MADUM » doit être modifié et donnez le MADUM dans le tableau suivant (en jaune le vieux MADUM):

MADUM

	Person()	Person(..)	C()	C(..)	placeOrder	setCustomerNumber	incrementRating				
name											
address											
rating							T				
trust											
customerNumber			C	C		T	O				
paidAmount			C	C							
order			C	C	T						

1.9 (1 point) Pour la classe Customer, combien de séquences de test faut-il écrire pour tester les transformers de

1.9.1 rating : we must reset the tranche as there are 4 transformers there will be 24 sequences but since we have 2 constructors we will have to write 48 test sequences

1.9.2 trust : nothing to do nothing change. Of course we must then test the other tranches.

1.10 (4 points) Identifiez les erreurs conceptuelles liées à l'attribut trust ou à d'autres attributs et expliquez si la technique du « MADUM » peut les révéler :

- 1) Constructors not always initialize the class attributes see for example `Pearson(...)` `Address()` or `Customer()` ; this is indeed a very poor practice.
- 2) `Person()` set trust to 0 but then `reset` set trust to 10
- 3) `Reset` suggest the cleanup of all but the nit just set trust to 10 ; even worse they compute trust as modulo %10 and thus it makes no sense to check if it is greater than 10
- 4) Getters (and setters) are missing