



POLYTECHNIQUE  
MONTRÉAL

## Questionnaire Mini-test 4

**LOG3430**

Sigle du cours

Questions	Points

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	20141
Professeur		Local	Téléphone
Giuliano Antoniol		C-630	
Jour	Date	Durée	Heures
Mercredi	12 Mars 2014	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 cellulaires, agendas électroniques ou téléavertisseurs sont interdits.
Directives particulières			
Toute documentation sera permise, ainsi que les calculatrices, à l'exception toutefois des téléphones cellulaires et de tout dispositif capable de connexion Internet.			

<b>Important</b>	Cet examen contient <input type="text" value="2"/> questions sur un total de <input type="text" value="8"/> 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.

**Question 1 (Max 7 points):**

Pour le programme suivant :

```
public class Bicycle {

    protected int gear;
    protected int speed;

    public Bicycle(int startSpeed, int startGear) {
        gear = startGear;
        speed = startSpeed;
    }

    public void setGear(int newValue) { gear = newValue; }

    public void applyBrake(int decrement) { speed -= decrement; }

    public void speedUp(int increment) { speed += increment; }

}

public class MountainBike extends Bicycle {

    protected int seatHeight;

    public MountainBike(int startHeight, int startSpeed, int startGear) {
        super(startSpeed, startGear);
        seatHeight = startHeight;
    }

    public void setSeatHeight(int newValue) { seatHeight = newValue; }

    public void fastBrake() { speed = 0; }

}
```

**Question 1.1 (1 point) :** Pour la classe Bicycle, complétez la table suivante pour le MaDUM, ajoutez des reporters si nécessaire (t transformer, c constructor, o other, et r reporter) ; voir l'exemple pour Item(n,p,a) qui est « constructor »; les données doivent être spécifiées dans le tableau suivant :

	Bicycle							
speed	c							
gear	c							

**Question 1.2 (1 point) :** Identifiez les tranches pour les attributs *speed* et *gear*; les données doivent être spécifiées dans le tableau suivant :

Attribute	Tranche							
speed								
gear								

**Question 1.3 (1 point) :** Pour les tranches *Tranche(speed)* et *Tranche(gear)*, donnez les séquences de test dans les tableaux suivants :

séquences de test pour <i>Tranche(gear)</i>					

séquences de test pour <i>Tranche(speed)</i>					

**Question 1.4 (1 point) :** Pour la classe MountainBike, complétez la table suivante pour le MaDUM, ajoutez des reporters si nécessaire (t transformer, c constructor, o other, et r reporter) ; les données doivent être spécifiées dans le tableau suivant :

speed										
gear										
seatHeight										

**Question 1.5 (1 point) :** Identifiez les tranches pour l'attribut seatHeight ; les données doivent être spécifiées dans le tableau suivant :

Attribute	Tranche			
seatHeight				

**Question 1.6 (1 point) :** Pour la tranche Tranche(seatHeight), donnez les séquences de test dans les tableaux suivants:

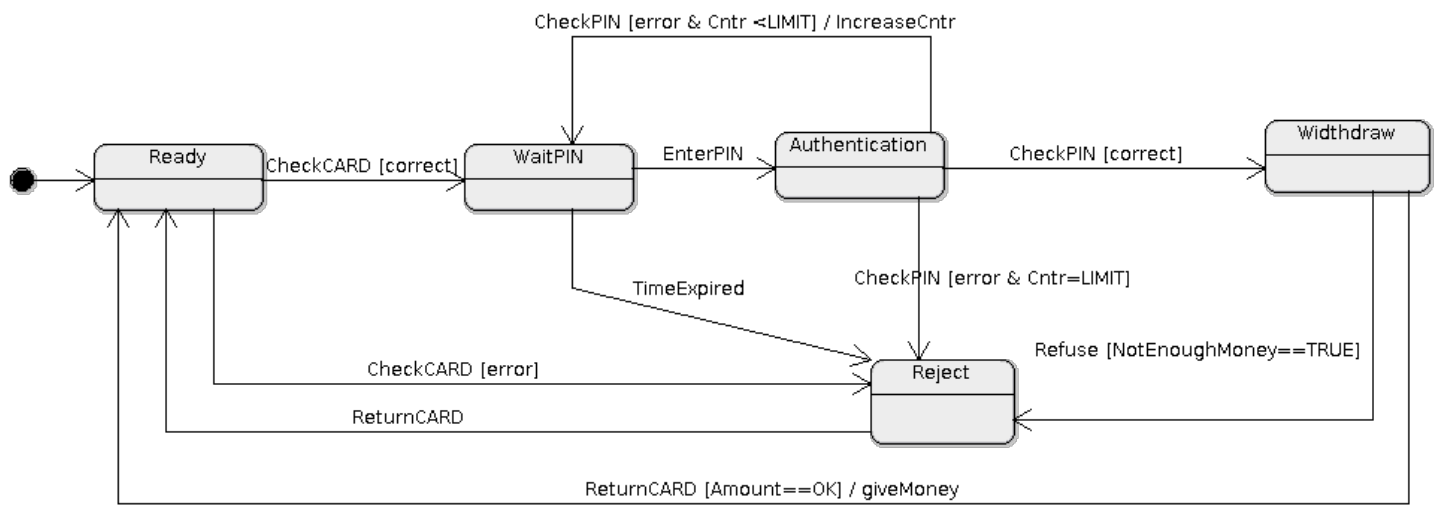
séquences du test pour Tranche(seatHeight)					

**Question 1.7 (1 point) :** Faut-il retester les tranches ? Si oui, donnez les séquences du test dans les tableaux suivants (justifiez la réponse!):

séquences de test							

**Question 2 (Max 13 points) :**

Pour le diagramme suivant :



**Question 2.1 (3 points) :** Écrivez 'arbre de transition selon la convention : S0=Ready, S1=WaitPIN, S2=Authentication, S3=Withdraw, S4=Reject :

---

**Question 2.2 (4 points) :** Décrivez les séquences de test. **Pour chaque cas de test, il faut justifier et expliquer si des conditions spéciales sont nécessaires ( !)**

**Question 2.3 (2 points) :** Pour chaque cas de test, donnez/décrivez l'oracle, séquence des états, et les conditions requises pour traverser le chemin. (Il n'est pas requis de compléter les cas de test en identifiant les valeurs des paramètres de la méthode).



**Question 2.4 (4 points) :** Est ce que l'arbre généré en réponse à la question 2.1 est unique ? Justifiez votre réponse. Si non, donnez un arbre (valide!) différent de l'arbre donné à la question 2.1.