



POLYTECHNIQUE  
MONTRÉAL

## Questionnaire Contrôle Périodique1

**LOG3430**

Sigle du cours

Identification de l'étudiant(e)		
Nom :	Prénom :	
Signature :	Matricule :	Groupe : 1

Sigle et titre du cours		Groupe	Trimestre
LOG3430 - Méthodes de test et de validation du logiciel		Tous	20171
Professeur		Local	Téléphone
Soumaya Medini		C-624	
Jour	Date	Durée	Heures
Mardi	24 janvier 2017	1 heure	

Documentation	Calculatrice	
<input type="checkbox"/> Aucune	<input type="checkbox"/> Aucune	Les cellulaires, agendas électroniques ou téléavertisseurs sont interdits.
<input checked="" type="checkbox"/> Toute	<input checked="" type="checkbox"/> Toutes	
<input checked="" type="checkbox"/> Voir directives particulières	<input type="checkbox"/> Non programmable	

Directives particulières
Toute documentation est permise, ainsi que les calculatrices, les ordinateurs portables et les tablettes électroniques. Vous n'avez toutefois pas le droit de vous connecter à Internet.

<b>Important</b>	Cet examen contient <input type="text" value="1"/> exercice et <input type="text" value="1"/> question sur un total de <input type="text" value="6"/> 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 – 15 points

Pour calculer le nombre de points d'inaptitude pour une infraction qui seront inscrits dans le dossier d'un conducteur, l'agent de police doit prendre en compte un nombre de facteurs importants. Dans un pays imaginaire, ce calcul est fait comme suit :

- En cas d'omission de se conformer à un panneau d'arrêt, 3 points d'inaptitude seront inscrits.
- En cas d'excès de vitesse :
  - o 11 à 30 km/h, 2 points d'inaptitude seront inscrits.
  - o Plus de 31km/h,
    - 10 points d'inaptitude seront inscrits si l'infraction est commise dans une zone de moins de 60 km/h;
    - 7 points d'inaptitude seront inscrits si l'infraction est commise dans une zone où la vitesse est limité entre 60 km/h et 100/h;
    - 4 points d'inaptitude seront inscrits si l'infraction est commise dans une zone de plus de 120 km/h.

Noter que dans ce pays imaginaire la limitation de vitesse maximale sur les routes est de 140km/h. Vous devez tester le logiciel qui permet de calculer les points d'inaptitude pour un conducteur dans ce pays imaginaire.

- 1) Faire l'analyse du problème selon la méthode des classes d'équivalence en considérant les classes des valeurs valides et invalides. (4 points)

(Les classes d'équivalence correspondante à des valeurs invalides sont en rouge.)

Infraction d'arrêt :

ia1:      Vrai  
ia2:      Faux

Infraction d'excès de vitesse :

iv1 : < 0  
iv2 : [0---11[  
iv3 : [11---30]  
iv4 : >31

Zone d'infraction de vitesse :

zone1: <0

zone2 : [0---60[

zone3 : [60---100]

zone4 : ]100---120] (La spécification n'étant pas complète, on peut décider de gérer cette zone comme étant invalide ou valide. Ici on prend le deuxième choix et on considère le même comportement que dans la zone3)

zone5 : [120---140]

zone6 : >140

**2) Concevoir le jeu de test selon le critère WECT. (3 points)**

WECT: 4 cas de tests (= max (2,3,4))

Cas de tests :

Infraction d'arrêt	Infraction de vitesse	Zone d'Infraction de vitesse	Sortie attendue
Vrai	5	30	3
Faux	15	60	2
Vrai	35	120	10
Faux	5	110	0

**3) Concevoir le jeu de tests selon le critère SRECT et donner le nombre de tests à concevoir. (4 points)**

SRECT: 48 cas de tests = 2\*4\*6

Infraction d'arrêt	Infraction de vitesse	Zone d'Infraction de vitesse	Sortie attendue
Vrai	-5	-5	ENTREE INVALIDE
Vrai	-5	40	ENTREE INVALIDE
Vrai	-5	70	ENTREE INVALIDE
Vrai	-5	110	ENTREE INVALIDE
Vrai	-5	130	ENTREE INVALIDE
Vrai	-5	150	ENTREE INVALIDE
Vrai	5	-5	ENTREE INVALIDE
Vrai	5	40	3
Vrai	5	70	3
Vrai	5	110	3
Vrai	5	130	3
Vrai	5	150	3
Vrai	20	-5	ENTREE INVALIDE

Vrai	20	40	5
Vrai	20	70	5
Vrai	20	110	5
Vrai	20	130	5
Vrai	20	150	5
Vrai	35	-5	ENTREE INVALIDE
Vrai	35	40	13
Vrai	35	70	10
Vrai	35	110	10
Vrai	35	130	7
Vrai	35	150	ENTREE INVALIDE
Faux	-5	-5	ENTREE INVALIDE
Faux	-5	40	ENTREE INVALIDE
Faux	-5	70	ENTREE INVALIDE
Faux	-5	110	ENTREE INVALIDE
Faux	-5	130	ENTREE INVALIDE
Faux	-5	150	ENTREE INVALIDE
Faux	5	-5	ENTREE INVALIDE
Faux	5	40	0
Faux	5	70	0
Faux	5	110	0
Faux	5	130	0
Faux	5	150	0
Faux	20	-5	ENTREE INVALIDE
Faux	20	40	2
Faux	20	70	2
Faux	20	110	2
Faux	20	130	2
Faux	20	150	2
Faux	35	-5	ENTREE INVALIDE
Faux	35	40	10
Faux	35	70	7
Faux	35	110	7
Faux	35	130	4
Faux	35	150	ENTREE INVALIDE

- 4) Pour la zone d'infraction de vitesse, proposer les valeurs pour l'analyse des valeurs limites. Notez qu'il n'est pas nécessaire de concevoir le jeu de tests, mais seulement d'énumérer les valeurs. (2 points)

0, 1, 30, 59, 60, 61, 80, 99, 100, 101, 110, 119, 120, 121, 130, 139, 140.

- 5) Discuter la qualité des tests et la confiance que vous allez avoir concernant le fonctionnement du logiciel si vous exécutez les tests conçus dans les questions précédentes. (2 points)

Si on analyse les valeurs possibles de sortie attendue (ENTREE INVALIDE, 0, 2, 3, 4,5, 7, 10, 13). On s'aperçoit que les cas de tests WRECT couvrent toutes les possibilités (c-à-d. toutes les valeurs possibles de la sortie). Ceci nous donne confiance que si les tests sont réussis, le logiciel fonctionne correctement suivant la spécification et il se comporte adéquatement avec des données invalides (c-à-d il est robuste). En revanche, pour beaucoup des cas de tests, la sortie est ENTREE INVALIDE (24 sur les 48). Ceci est dû au fait que la méthode des classes d'équivalence considère que les entrées sont indépendantes et combine donc les classes d'entrée invalides avec toutes les classes des autres entrées.

### Question 1 – 5 points (1 point par sous-question)

- 1.1 Si nous avons la possibilité d'exécuter des tests unitaires d'une manière exhaustive nous, n'aurons pas besoin de tests d'acceptation. (1 point)

- |       |                          |                                     |
|-------|--------------------------|-------------------------------------|
| 1.1.1 | Vrai .....               | <input type="checkbox"/>            |
| 1.1.2 | Faux.....                | <input checked="" type="checkbox"/> |
| 1.1.3 | Question mal posée ..... | <input type="checkbox"/>            |
| 1.1.4 | Ça dépend .....          | <input type="checkbox"/>            |

**Justification: Faux** car les tests unitaires peuvent révéler des fautes de fonctionnement d'un module mais pas des fautes d'interaction entre les modules.

- 1.2 Les défauts introduits dans l'implémentation du logiciel peuvent être découverts par les tests boîte noire. (1 point)

- |       |                     |                                     |
|-------|---------------------|-------------------------------------|
| 1.2.1 | Vrai .....          | <input type="checkbox"/>            |
| 1.2.2 | Faux.....           | <input checked="" type="checkbox"/> |
| 1.2.3 | Ça dépend .....     | <input type="checkbox"/>            |
| 1.2.4 | Je ne sais pas..... | <input type="checkbox"/>            |

**Justification: Faux** car les tests boîte noire permettent de détecter les fautes introduits dans la spécification.

**1.3** Les tests boîte blanche permettent de trouver plus de fautes que les tests boîte noire. (1 point)

- 1.3.1 Vrai ..... ☐
- 1.3.2 Faux..... ☒
- 1.3.3 Ça dépend ..... ☐
- 1.3.4 Je ne sais pas ..... ☐

**Justification: Faux** car comme le types d’erreurs visés par les 2 familles de stratégies sont différentes on ne peut pas affirmer que l’une ou l’autre est plus efficace.

**1.4** Les tests de robustesse s’assurent que le programme est correct et que le calcul est fait comme la spécification le demande. (1 point)

- 1.4.1 Vrai ..... ☐
- 1.4.2 Faux..... ☒
- 1.4.3 Ça dépend ..... ☐
- 1.4.4 Je ne sais pas ..... ☐

**Justification: Faux** car les tests de robustesse se concentrent sur le comportement du programme dans des conditions anormales.

**1.5** L’analyse des valeurs limites est bien adaptée pour tester un programme qui prend en entrée des chaînes de caractères, car elle nous permettra de révéler des fautes de conception. (1 point)

- 1.5.1 Vrai ..... ☐
- 1.5.2 Faux..... ☒
- 1.5.3 Ça dépend ..... ☐
- 1.5.4 Je ne sais pas ..... ☐

**Justification: Faux** car cette technique est adaptée pour des variables physiques (telles qu’une température, une longueur, etc.).