(ECE453/CS447/ECE653/CS647/SE465)
Software Testing, Quality Assurance & Maintenance:
Assignment/Lab 3 (60 Points)

Instructor: Lin Tan

March 12, 2013

Due: 5:00 PM, Thursday, March 28, 2013
Submit: An electronic copy on LEARN

Please download "HashTable.java"for Q3 from LEARN.

I expect each of you to do the assignment independently. I will follow UW's Policy 71 if I discover any cases of plagiarism.

Submission Instructions:

Please read the following instructions carefully. **If you do not follow the instructions, you may be penalized up to 5 points.** Illegible answers receive no point.

Go to "Dropbox" $-\rangle$ "Submit A3" on LEARN. Submit only one file in .tar.gz format. Please name your file

<FirstName>-<LastName>-<StudentNo>.tar.gz

For example, use John-Smith-12345678.tar.gz if you are John Smith with student No 12345678. *You must include your file name as part of your submission title.* The .tar.gz file should contain the following items:

- a single pdf file "*a3_sub.pdf*".
  You must include a cover page that contains your full name, 8-digit student No, the class number (one of ECE453, CS447, ECE653, CS647, & SE465), and your uwaterloo email address.

You can submit multiple times. After submission, you can view your submissions to make sure you have uploaded the right files/versions.

Question	TA in Charge
1	Jinqiu
2	Lei
3	Pei

Question 1 (20 points)

For predicate $p = (a \wedge b) \vee (b \wedge c) \vee (a \wedge c)$, answer the following questions:

- (a) Compute and simplify the conditions under which each of the clauses determines predicate p. The condition must only use $\wedge$, $\neg$, and $\vee$. Complete the table. (8 points)
- (b) Identify all pairs of rows from your table that satisfy GACC with respect to each clause. (4 points)
- (c) Identify all pairs of rows from your table that satisfy CACC with respect to each clause. (4 points)
- (d) Identify all pairs of rows from your table that satisfy RACC with respect to each clause. (4 points)

	a	b	c	p	p_a	p_b	p_c
1	T	T	T				
2	T	T	F				
3	T	F	T				
4	T	F	F				
5	F	T	T				
6	F	T	F				
7	F	F	T				
8	F	F	F				

Question 2 (20 points)

Answer the following questions for the method intersection() below:

```
public Set intersection (Set s1, Set s2)
// Effects: If s1 or s2 is null, throw NullPointerException
// else return a (non null) Set equal to the intersection of Sets s1 and s2

Characteristic: Size of s2
- s2 has even number of elements (in other words, the size of s2 is an even number)
- s2 has odd number of elements (in other words, the size of s2 is an odd number)

Characteristic: Relation between s1 and s2
- the size of s1 is not bigger than the size of s2
- the size of s2 is not bigger than the size of s1
```

- (a) Does the partition "Size of s2" satisfy the completeness property? If not, give a value for s2 that does not fit in any block. (4 points)
- (b) Does the partition "Size of s2" satisfy the disjointness property? If not, give a value for s2 that fits in more than one block. (4 points)
- (c) Does the partition "Relation between s1 and s2" satisfy the completeness property? If not, give a pair of values for s1 and s2 that does not fit in any block. (4 points)
- (d) Does the partition "Relation between s1 and s2" satisfy the disjointness property? If not, give a pair of values for s1 and s2 that fits in more than one block. (4 points)
- (e) If the "Base Choice" criterion were applied to the two partitions (exactly as written), how many test requirements would result? (4 points)

Question 3 (20 points)

In this question, you will criticize and improve an existing bug report in Mozilla and write your own bug reports.

- (a) Read the Mozilla bug report 112785 (`https://bugzilla.mozilla.org/show_bug.cgi?id=112785`). What are the problems of the initial bug report (the "Title" and the "Description")? Identify four problems. How would you improve this bug report? (10 points)
- (b) The program *HashTable.java* is an implementation of hash table using linear open addressing and division in Java. This program has a bug, because the `put` function will NOT update the element associated with the given key if an entry with the same key is put. However, this hash table implementation should update the element even if the element's key already exists in the hash table (see the comment in the `put` function).

  Write a good bug report for this bug using the Firefox bug report format. (10 points)