# Spring 2017 Qualifying Exam – Languages

Your Test ID Number: _____

## Instructions

Write your test id number above and on each page of your answers. This exam is closed book and closed notes. You must pass both parts to pass the test.

## Part 1: CS 440 [50 points]

(1) (30 points) For each pair of programming language terminologies in the following, give the main difference between them and the main advantage of one over the other. For example, in the first pair, you will give the main difference between an ambiguous grammar and an unambiguous grammar, an advantage of the ambiguous grammar, and an advantage of the unambiguous grammar.

a) (6 points) Ambiguous grammar VS Unambiguous grammar

b) (6 points) Call-by-Value VS Call-by-References

c) (6 points) Deterministic Finite State Automata and Non-Deterministic Finite State Automata

d) (6 points) Declarative programming vs imperative programming

e) (6 points) Pointers (as in C) vs references (as in Java)

## (2) (10 points)

(a) (2 points) Describe regular grammar and context-free grammar with examples.

(b) (3 points) Consider the following grammar, state whether it is ambiguous. If it is, give an equivalent unambiguous grammar.

If it is not, state all the precedence and associativity enforced by the grammar.

In the grammar, '∧', '∨', '∩', '∪', '(', ')', '+', and '*' are all terminal symbols. Assume *Id* generates identifiers.

$$E ::= E \cap F \mid E \cup F \mid E \vee F \mid E \wedge F \mid F$$

$F$ ::= $Id$ | ($E$)

(c) (2 points) What is the differences between Parse tree and Abstract Syntax tree?

(d) (3 points) Consider the following grammar:

$P$ ::= $id$ | $P \vee P$ | $P \wedge P$ | $\neg P$ | ($P$)
$id$ ::= $p$ | $q$ | $r$ | $s$

where the set of terminals symbols is {$\vee$, $\wedge$, $\neg$, (, ), p, q, r, s} and the start symbol is $P$. Give leftmost derivation for the following expression

($p \wedge q$) $\vee \neg$($p \wedge q$)

## (3) (10 points)

Java has *checked exceptions* but C# does not.

(a) (3 points) What is a *checked exception*? Give an example code fragment that will pass the exception checking and one that will not.

(b) (4 points) Give a reason for and a reason against checked exceptions.

(c) (3 points) Suppose that you are the language designer who must make the decision of whether to include checked exceptions, what will be your position and why?

## Part 2: CS 536 [50 points]

1. (4 points) Calculate wlp(**if** x > 0 → x := x+1 □ y > 0 → y := y+1 **fi**, 0 < x < y). Show your work.
2. (8 points) Calculate wp(b[x] := b[y]; b[y] := b[z], b[x] ≤ b[y]) Show your steps; you should use syntactic and logical transformations to simplify your answer.
3. (8 points) Calculate sp(x < y, x := f(x, y); x := f(x, z)). Show your work

4. (14 points) Complete the program below by adding a final postcondition. Give a full outline for total correctness of your program (including invariant and bound function). Feel free to write ^ for infix exponentiation.

   $\{n = n_o \wedge n \geq 1\}$

   x := k; n := n-1;

   **while** n > 0 **do**

      x := x * k; n := n-1

   **od**

5. (8 points) What are the interference freedom checks for {p1} S1; {q1} T1 {r1} and {p2} S2 {q2} (assuming the statements are atomic)?

6. (8 points) List all the deadlock-freedom checks for the following parallel program outline.

   [ {p1} S1; {q1} **await** B1 **do** {r1} T1 {s1} **end** {s1} U1; {t1}

   || {p2} **await** B2 **do** S2 **end**; {q2} **await** C2 **do** {r2} T2 {s2} **end**; U2 {t2} ].