Programming Language Qualifying Exam
Spring 2007

Answer any five of the following six problems. You may attempt all six, and we will use the five highest scoring for your grade.

1. Languages and Compilation

   (a) Programming languages can be grouped according to the paradigm used. The four most well known are *imperative*, *object-oriented*, *functional*, and *logic*. Explain each of these paradigms by giving a list of their prominent features, and an example of the kind of problem such a language would be well-suited to solve.

   (b) C++ and Java are very similar languages. What is the point of having both of them? Give two advantages that C++ has over Java, and two advantages that Java has over C++.

2. Abstraction

   (a) What are abstract data-types? Why are they important?

   (b) There is a tool called i`yacc` which implements a parser-generator. It works by taking a file containing a specification of a grammar, say `foo.grm` and outputting an implementation of that grammar in a file, e.g., `foo.c`. This file is then compiled with a C compiler and linked to the application, thereby providing a parser.

   If you open `foo.c`, you will see that no attempt is made to hide data. Everything is implemented via arrays and pointers, with no thought about abstraction. Give two reasons why is this acceptable.

3. Grammars

   Consider the following grammar:

$$\begin{array}{rcl} S & \rightarrow & a\ E\ a \\ E & \rightarrow & E\ x\ a \\ & | & a\ x\ E \\ & | & x \end{array}$$

   (a) Construct the Characteristic Finite State Machine for the above grammar.

   (b) Is the above grammar LL? Why or why not?

   (c) Is the above grammar ambiguous? Give a proof with your answer.

4. Weakest Precondition

   (a) Define *weakest precondition*.

   (b) Given program S_1, and S_2, and predicates P, Q, and R, suppose that $wp(S_1, R) = Q$, and $wp(S_2, P) = R$.

i. What combination of programs and initial state x will ensure that P is true after the programs are run?

ii. For the above situation, suppose x does not satisfy those initial conditions. Is it possible that P could be true after the programs are run?

iii. What does it mean if $P \equiv true$?

iv. What does it mean if $Q \equiv true$?

(c) Consider the following program S. Let the postcondition $R \equiv m = min(x, y)$. Determine formally the conditions under which this program returns the correct answer.

```
if x > y then m := y;
if x < y then m := x;
```

5. Loop Verification

(a) In order to verify the correct operation of a loop, you need to check five formulas. What are they?

(b) Formally prove or disprove the following loop. The postcondition is $s = min_{i=0}^{n-1} A[i]$. You will need to determine the loop invariant.

```
s := 0;
i := 0;
do i < n -> s,i := min(s,a[i]), i + 1
od
```

6. Parallelism Suppose you have two programs $A \equiv A_1; A_2; A_3$ and $B \equiv B_1; B_2; B_3$. Both A and B are assumed to be correct when run individually, but now we want to run them in parallel.

(a) What is *interleaving semantics*? Give an illustration using the above programs.

(b) Suppose A and B communicate via shared variables. What needs to be true in order for correctness to be maintained?

(c) Suppose A and B communicate using message passing. What needs to be true in order for correctness to be maintained?

(d) What is the difference between deadlock and divergence?