Programming Language Qualifying Exam
Spring 2008

Answer any five of the following six problems.

1. **Languages and Compilation**

   (a) Compiled programs run faster than interpreters. Why, then, do we still have interpreters?

   (b) Suppose there is a class A, with two classes B and C, both of which inherit from A. All three classes define the method `foo`. Suppose also that we declare an array of type A and fill it randomly with objects of type A, B, and C. Then, for each item in the array, we call a method `foo`. Even though the array is of type A, the appropriate version of `foo` is called. Explain the mechanism by which this happens.

2. **Abstraction**

   (a) What is an abstract data-type? What advantages are there to using an abstract data type?

   (b) Programming team A implemented a queue by writing a new class, providing the three queue methods, and calling the list class internally. Programming team B implemented a queue by adding the three queue methods to the list class, and instructed their programmers to use the list class whenever a queue was needed. Which team made the right decision, and why? What is the disadvantage of the wrong team's approach?

3. **Grammars**

   Consider the following grammar:

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

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

   (b) Convert the above grammar to an LL grammar (or explain why it is already LL).

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

4. **Weakest Precondition**

   (a) Give the definition of *weakest precondition*.

   (b) Suppose $WP(S, Q) = P$. Is it possible that there could be some P' such that $P' \Rightarrow P$? Is it possible that there could be some P'' such that $P \Rightarrow P''$?

   (c) Give the definition of WP for an `if` statement.

   (d) Construct a simple program S such that $WP(S, P) \cup WP(S, Q) \neq WP(S, P \cup Q)$, for some assertions P and Q.

5. **Loop Verification**

   (a) To verify a loop, you need to solve five equations. List each equation and give a one sentence description of its role in the verification.

   (b) We want a program that, given an array $A[0..N]$, sets the integer m to be the maximum element of the array.

      i. Write a specification for your program by giving a precondition Q, postcondition R, and loop invariant P.

      ii. Write the program, and formally verify it.

6. **Application**

   Consider the following game. A jar contains a non-zero, finite number of coins. The coins are of two types, W and L. A Student plays by non-deterministically removing five coins from the jar. If there are fewer than five coins, then Student takes all of them. Next, Student looks at the coins that were just drawn, and decides either to put one of them back into the jar, or to do nothing. All coins not returned to the jar are then discarded. If there is one coin left in the jar, the game terminates; otherwise, Student can play another round. Student wins the game if the game terminates with one coin of type W in the jar, and loses if the game terminates with one coin of type L in the jar.

   (a) Write the strategy for the game. What type of coin should the student put back as a function of the five coins that were picked up?

   (b) Prove that the game always terminates. Add **ONLY** the assertions and invariants necessary to help you reason about termination.

   (c) Now add the assertions and invariants necessary to prove that Student will always win.

   (d) Finally, add the assertions and invariants necessary to prove that Student never fails. (Hint: this is not the same thing as the previous question!)