# Programming Language Qualifying Exam
## Fall 2007

Answer the following five problems.

1. Languages and Compilation

   (a) What is the difference between an interpreter and a compiler?

   (b) Java can be considered both an interpreter and a compiler. How is that so? What is the advantage of designing the language that way?

   (c) Languages like Ruby, BASIC, Python, Lisp, and Php are dynamically typed. Languages like Fortran, C, OCaml, and Pascal are statically typed. What is the difference between static typing and dynamic typing? Give an advantage for each model and a circumstance under which you would want to use such a language.

   (d) Objects in C++ and Java can behave both as statically typed objects and dynamically typed objects. How is that accomplished?

2. Abstraction

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

   (b) One of the major changes from C++ to Java was the replacement of pointers with references. Give at two advantages of having made this change.

   (c) Give a situation in which using C-style pointers rather than Java-style references is preferable from a software engineering standpoint.

3. Grammars

   Consider the following grammar:

$$
\begin{aligned}
S &\rightarrow x\ E \\
E &\rightarrow E\ y\ E \\
&\mid\ a\ b
\end{aligned}
$$

   (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) What advantage results from a grammar being LL?

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

4. Weakest Precondition

   (a) Define *weakest precondition.*

   (b) Given deterministic program $S_1$, and predicates $P_1$, $P_2$, $Q$, and $R$, suppose that $wp(S_1, R) = Q$, and that $P_1 \rightarrow Q$, and that $Q \rightarrow P_2$.

      i. Does $\{P_1\}$ $S_1$ $\{R\}$ hold?
      ii. Does $\{P_2\}$ $S_1$ $\{R\}$ hold?

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

```
if x > y then x := y;
if x < y then y := 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) Fix the bug in the following program (if there is one), and formally prove the result. The postcondition is $s = \Pi_{i=0}^{n-1} a[i]$. Remember that $\Pi$ indicates product. You will need to determine the loop invariant.

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

   (c) Writing proofs can be a lot of work. Why not just use testing instead of formal methods to prove programs correct?