The exam will be on Thursday, 10 December from 9 a.m. to noon. It is closed book, closed notes, with one exception noted below. You may not consult a computer or any other device during the exam.

You may bring one 8-1/2 by 11 sheet of notes that you may consult during the exam.

The exam will cover the entire semester, but will focus primarily on the topics listed below.

- Fundamental terminology—things like “binding,” “scope,” “dynamic,” “static,” “polymorphism,” ”composite types,” “right-associative,” “side effect,” “row-major,” and a number of others, with particular focus on more recent topics. This is *not* a complete list of terms! *Study the slides!*

- Particular features of particular languages, most particularly:
  - different scoping rules, with examples in Java, C, JavaScript, Haskell. This includes block scoping, function-level scoping, and mechanisms such as the “let” scoping construct in Haskell
  - variations in parameter passing (value vs. reference, default values, named arguments, normal, applicative, and lazy order, etc.), with examples in C, Java, Haskell, Python
  - different method binding rules in object-oriented languages, with examples from Java and C++
  - common functional programming operations such a currying, composition, “map,” “reduce,” etc., with examples in Lisp and Haskell

- Compilation, interpretation, and their relationship to the language elements we’ve been looking at all semester

- Basic notions of binding and scoping plus new information from more recent chapters (e.g., static and dynamic method binding)

- Basic notions of type systems and common data types (arrays, pointers, and other composite types) plus new information from more recent chapters (e.g., examples of Haskell’s typing system)

- Basic notions of functions and procedures, including newer material (e.g., definition of “closure”)

- Basic notions of object-oriented programs, with examples from Java, Python, and C++
• Broad notions of the concepts of logic programming (as explained on the slides)

Questions will be aimed at basic understanding of concepts and the ability to apply them in concrete examples. Question types will include:

• Short answer (may occasionally require writing simple code)
• True/false
• Multiple choice

You will not be asked to write whole programs; however, you have been exposed to a number of concepts through languages other than Java (in labs, in class) and you will be expected to recognize features of such languages when they were highlighted in class or in one of the lab assignments (e.g., the “map” operator in Haskell or the “multiple inheritance” feature in C++).

See the review sheets for exams 1 and 2 for many samples of exam questions. In addition, here are a few more examples of the kinds of questions that might be asked about more recent topics.

1. Given the following Prolog database:

   language(java).
   language(haskell).
   loves(mary,java).
   loves(mary,haskell).
   loves(john,pizza).
   programmer(X) :- loves(X,Y),language(Y).

   (a) What values of $X$ (if any) satisfy the query “loves(mary,X).”?
   (b) What values of $X$ (if any) satisfy the query “loves(X,haskell)”?
   (c) Is the following query true? “programmer(john)”?
   (d) What values of $X$ (if any) satisfy the query “programmer(X)”?

2. Define the term closure.

3. What are the characteristics of object-oriented programming?

4. What are the values of the following Lisp expressions?

   (a) $(+ (* 2 3) (- 4 3))$
   (b) (list 'a 'b 'c)
   (c) (rest '(1 2 3 4 5))
   (d) (first (rest '(1 2 3 4 5)))

5. In Haskell, we type the command “:t f” and see:

   $f :: \text{Num} \ a \Rightarrow \ a \rightarrow \ a$
What is this showing us?

(a) the value of function $t$ when applied to argument $f$
(b) the value of function $f$ when applied to argument $\text{Num}$
(c) the type of function $f$
(d) nothing—this is an attempt to trick you with a nonsense question

6. Define side effect and give an example.