Open Book
You may bring *Object-Oriented Common Lisp* by Stephen Slade, and the handouts: the Logic Programming chapter from Norvig and the two Search chapters from Russell and Norvig.

Topics
- Evaluation: order of evaluation, special forms vs ordinary evaluation
- Variables and bindings: bound vs fbound, local vs global lexical environments, let, let*, labels
- Control structure: if, cond, unless, when
- Iterator structures: do, do*, loop (simple), dolist,
- Data structures: lists, box and pointer diagrams, car/cdr representation, structures, hash tables
- Functions: defining and using function, functions as arguments, creating and using closures, recursion, tail recursion
- Strings: string and character data, format, concatenate
- I/O: opening and closing files, with-open-file
- Prolog: symbolic pattern matching, pattern variables, unification, Prolog rule and assertion definition, relational definitions (member, length)
- Macros: difference between macro expansion and function evaluation, backquote and comma syntax, read macros
- Search: representing problems as search spaces, problem domain decomposition in GPS (states and operators), search algorithms (dfs, bfs, beam search, heuristic search, A*)
- CLOS: classes and method, generic functions vs message passing, method combination
- Examples: iterators, symbolic mathematics, Eliza

Questions to expect
- Converting a recursive function to a tail-recursive one.
- Tracing a Lisp function and determining its output and/or side-effects
- Defining a CLOS class and associated methods
- Giving the expansion of a simple macro
- Writing a function that returns a closure
- Converting a recursive function into an iterative one
- Given some assertions, write a Prolog rule that will prove a consequent
- Conceptual questions