CSCI 330 (Programming Languages)
Dr. Schwartz
Review Sheet for Exam 2

This is intended as a guideline for studying for the second exam… but only as a guideline! I wouldn't have covered something if I didn't think it was important. If you are wondering about a topic and you don't see it here, ask me!

Approximate point distribution (30-40% Chapter 5, 30-40% Chapter 6, 15-25% Chapter 7, 5-10% OCaml)

Sample Types of Questions

- Short answer, multiple choice and true/false, fill-in-the-blank
- Problem solving
    - Static vs. dynamic scope
    - Referencing environments
    - Analyze effects of operand evaluation order with functional side effects
    - Analyze effects of short-circuit evaluation

Be sure to look at and think about the questions at the end of the chapters. Look at the "review questions" and the "problem sets" (not the programming exercises).

Yes, yes, I know that the topics are a list of what we covered, but they're a sanity checklist for you. ☺

Chapter 5 (Names, Bindings and Scope)

- Names – design issues (form, case sensitivity, keywords vs reserved words)
- Variables – name, address, type, value, lifetime, scope
- Type binding – static vs dynamic, type inference
- Storage binding – static, stack-dynamic, explicit heap dynamic, implicit heap dynamic
- Scope – static vs dynamic, referencing environment
- Scope vs lifetime

Chapter 6 (Data Types)

- Definitions: Data type, descriptor, object
- Primitive Data Types (what makes a type primitive?)
- Strings: Design issues, language options, static vs. dynamic length, …
- Ordinal types, enumerations, subranges
- Arrays
    - subscript types
    - subscript binding (static, fixed stack-dynamic, stack-dynamic, fixed heap-dynamic, heap-dynamic)
    - operations
    - rectangular and jagged (slices)
    - descriptors
    - associative arrays
- Record Types: design issues, operations, references
- Unions: design issues, discriminated vs. free
- Pointers: design issues, operations, problems (and programming language solutions), pointer arithmetic
- Heap management: single-size cells vs variable size, mark-sweep vs reference counting for garbage collection

- Type checking: strongly typed, name type equivalence, structure type equivalence

Chapter 7 (Expressions and Assignment Statements)

- Arithmetic expressions: design issues
  - operators (unary, binary, ternary)
  - operator precedence and associativity
  - operand evaluation order
  - side effects
  - overloaded operators
  - type conversions: narrowing/widening, mixed-mode (coercion), explicit
  - Boolean expressions, short-circuit
- Assignment statements
  - Conditional assignment
  - Compound operators
  - Assignments as expressions
  - List assignments
  - Mixed-mode assignments

OCaml

- Writing and understanding OCaml code
  - Knowledge of constructs used in Labs 2 and 3
  - Writing tail recursive functions
  - Higher-order functions
  - Pattern matching
  - Tuples