**CS 291 Final Exam Terms and Concepts**

**Hein Section 6.2 Propositional Calculus**

* Be familiar with *truth tables* and know how to use them to show the truth value of statements.
* Know what makes a statement a well-formed formula (wff).
* Understand the hierarchy of evaluation for the logical connectives and be able to unambiguously interpret wffs.
* Know how to show that two wffs are *logically equivalent* by doing a step-by-step proof from one form to the other.
* You should be familiar with the basic equivalences from Figure 6.6 on page 404.
* Understand what it means for a wff to be a *tautology*, a *contradiction* or a *contingency* and be able to determine which of these any wff is using Quine's method.
* Be familiar with Conjunctive Normal Form (CNF) and Disjunctive Normal Form (DNF). Be able to turn a wff into either form using equivalences and using truth tables.

**Hein Section 6.3 Formal Reasoning**

* Know how to do proofs using natural deduction.
* You will have access to the Proof Rules on page 422 of your book. Know how to use them to do step-by-step proofs where each step is justified by previous steps and proof rules.
* This includes being able to do nested *Conditional Proofs* (CP) and *Indirect Proofs* (IP).

**Hein Section 6.4 Formal Axiom Systems**

* Know the definitions of *soundness* and *completeness* and how they differ.

**Hein Section 7.1 First-Order Predicate Calculus**

* Understand *predicates*, *existential quantifiers* and *universal quantifiers* over a *domain*.
* Know what a well-formed formula (wff) is and all the standard terminology associated with this idea.
* Be familiar with *scope* of quantified variables and how to distinguish *bound* from *free* variables.
* Know the concepts of *valid*, *unsatisfiable* and *satisfiable* as applied to wffs.
* Know what *universal* and *existential closures* are.

**Hein Section 7.2 Equivalent Formulas**

* Be familiar with the concept of *logical equivalence*.
* Be able to do various manipulations to show that one logical form is equivalent to another logical form.
* Know about *prenex normal form* and how to to through the steps to put a wff into this form.
* Beyond this, be able to put wffs into *disjunctive normal form* and *conjunctive normal form*.
* Be able to formalize English sentences into wffs and find natural sounding English sentences that are equivalent to wffs.

**Hein Section 7.3 Formal Proofs in Predicate Calculus**

* Know how to do formal conditional proofs in FOL using natural deduction.
* Be familiar with *universal instantiation (UI)*, *existential generalization (EG)*, *existential instantiation (EI)*, and *universal generalization (UG)*.
* Be able to take English sentences, translate them into formal wffs and go through the process of doing a formal proof.

**Hein Section 7.4 Equality**

* There are various proofs in this section that I won't expect you do do much with.
* You should know the *Equal for Equals (EE)* rule well enough to use it in proofs.
* You should extend your ability to formalize English sentences in FOL to include aspects of equality, as in the homework problems.

**Hein Section 8.1 Program Correctness**

* You should understand the {P} S {Q} syntax used in program correctness proofs.
* Know the *Assignment Axiom* and the *Consequence Rules* and how to use them in proofs.
* Be familiar with the *Composition Rule* and the *If-Then Rule* and *If-Then-Else Rule* and how to use them in proofs. If I expected you to use any of these three rules in proofs, I would give them to you on the exam.
* I do **not** expect you to be able to do correctness proofs with the *While Rule* or any of the array assignment stuff or program termination proofs.

**Hein Section 8.2 Higher-Order Logics**

* Nothing from this section.

**Hein Section 8.3 Automatic Reasoning**

* I expect you to be able to take logical wffs and put them in *clausal form*, including *Skolemization* as necessary. This includes being able to carefully apply *Skolem's Rule*.
* You should be able to do *resolution proofs*, both with propositional logic clauses and first-order logic clauses.
* Understand *substitution* and *unification* and how they are used in resolution proofs.
* Be able to use *Robinson's Unification Algorithm* to find *most general unifiers (mgus)*.
* To restate, be able to do full blown first-order logic resolution proofs from beginning to end.

**Hein Section 11.1 Regular Languages**

* Given a description of a language, be able to find a regular expression for it.
* Know how to do simple manipulations to show that two regular expressions are equivalent.

**Hein Section 11.2 Finite Automata**

* Given a DFA or NFA, be able to write down the transition table.
* Given a regular expression, be able to construct a DFA or NFA for recognizing that language.
* Given an FA, be able to find an equivalent regular expression.
* Be familiar with the Prolog code for representing and reasoning with FAs.
* Understand Mealy Machines and Moore Machines and be able to show what their output would be given an input string.

**Hein Section 11.3 Constructing Efficient Finite Automata**

* Be able to find the *lambda closure* of an NFA state or set of states.
* Know how to convert an NFA into an equivalent DFA.

**Hein Section 11.4 Regular Language Topics**

* Know how to use the *Pumping Lemma for Regular Languages* to show that particular languages are not regular.

**Hein Section 3.3 Grammars**

* Know how to do derivations given a grammar.
* Be able to find a grammar for a language given a description of the language.
* Understand how to show that a grammar is ambiguous by finding two parse trees for the same sentence.

**Hein Section 11.5 Context-Free Languages**

* Be able to find context-free grammars for languages.

**Hein Section 11.6 Pushdown Automata**

* Know what a pushdown automata is.
* Know how to construct a pushdown automaton for a given context-free language.
* Be aware that there is a version of the pumping lemma for showing that languages are not context-free.

**Hein Section 12.1 Turing Machines**

* Be able to describe the Turing Machine model of computation.
* Understand the concept of a *Universal Turing Machine* and why it is so important.

**Hein Section 12.2 The Church-Turing Thesis**

* Know what the Church-Turing thesis is and be able to give a brief explanation of it.

**Hein Section 12.3 Computability**

* Know what a *decision problem* is and what it means for a decision problem to be decidable or undecidable or partially decidable.
* Be able to explain what the *Halting Problem* is and approximately describe why it is undecidable.