```
Topics to Study for Exam 3


Chapter 6

intermediate code representations
static versus dynamic type checking
equivalence of type expressions
coercions, overloading, and polymorphism
translation of boolean expressions
backpatching and translation of flow-of-control constructs
translation of record declarations
translation of switch statements
translation of array references


Chapter 5

syntax directed definitions
synthesized and inherited attributes
dependency graphs
L-attributed definitions
translation schemes
syntax-directed construction of syntax trees and DAGs
syntax-directed translation with YACC


Chapter 7

activation records
call graphs
typical actions during calls and returns
storage allocation strategies
   static, run-time stack, heap
calling convention issues
heap storage reclamation strategies
access to nonlocal names
establishing access links
displays


Chapter 8

register assignment
instruction selection
implementation of conditional branches
SPARC addressing modes
run-time stack management
evaluation of arguments
code-generator generators


Chapter 9

types of compiler optimizations
   function call optimizations
      inlining
      cloning
      tail call elimination
      function memoization
   loop optimizations
      invariant code motion
      loop strength reduction
      basic induction variable elimination
      unrolling
      collasping
      fusion
      software pipelining
   memory access optimizations
```



```
      register allocation
      array padding
      scalar replacement
      loop interchange
      prefetching
   control flow optimizations
      branch chaining
      reversing branches
      code positioning
      loop inversion
      useless jump elimination
      unreachable code elimination
   data flow optimizations
      common subexpression elimination
      partial redundancy elimination
      dead assignment elimination
      evaluation order determination
      recurrence elimination
   machine specific optimizations
      instruction scheduling
      filling delay slots
      exploiting instruction-level parallelism
      peephole optimization
constructing a control flow graph
optimizations before and after code generation
instruction selection optimization


Chapter 10

instruction pipelining
data dependences
   true dependences
   antidependences
   output dependences
data dependence analysis for memory references
   array data dependence analysis
   points-to analysis
representing data dependences
eliminating false dependences
scheduling for a multi-issue processor
global instruction scheduling
increasing ILP
   loop unrolling
   software pipelining
```