Plan for today

Final review in the context of code generation for short circuiting
- type checking
- translation to IR Trees
- tree pattern matching and temp assignment
- code generation

Short Circuiting

Short circuited AND (&&)

```java
if (p < 10 && 2 < p) {
    x = 7;
} else {
    x = 22;
}
```

Changing the Translation

**Translate::outAAndExp (OLD way)**
- stmtList.addAll( stmts from l_exp )
- stmtList.addAll( stmts from r_exp )
- create new Tree.ExpBINOP(AND) and hook in Tree.Exp for l_exp and r_exp

**Translate::outAAndExp (for short circuiting)**
- stmtList.addAll( stmts from l_exp )
- stmtList.addAll( the following
  - StmCJUMP(EQ, Tree.Exp for l_exp, ExpCONST 1, L4, L3)
  - StmLABEL L3
  - StmMOVE( ExpTemp t35, ExpCONST 0 )
  - StmJUMP L5
  - StmLABEL L5
- stmtList.addAll( stmts from r_exp )
- stmtList.addAll( the following
  - StmMOVE( ExpTemp t35, Tree.Exp for r_exp )
  - StmJUMP L5
  - StmLABEL L5
- Tree.Exp for AndExp is ExpTEMP(t35)