

# 6.035 Info Session 2: Semantic Checker

Fall 2016

# Team Repositories Created

Java

Scala

Haskell

---

Antidaephibiax

Zeta Psience

FP

Robert'); DROP TABLE  
STUDENTS; --

Harambe's Compiler

We're Screwed!?!?

# Timeline

<b>Project</b>	<b>Points</b>	<b>Due</b>
Scanner-Parser		
<b>Semantic Checker</b>		March 1 (ungraded feedback)
Code Generator	25	March 18
Data-flow Analysis		
Optimizer	45	

# Components

**Parser:** from project 1

**Abstract syntax tree**

**High-level IR:** simplifies subsequent analysis  
design carefully

**Symbol tables:** hold meta-information

**Semantic checks**

# Building AST in Parser

`program: (s:stmt)* {#program = #([PROGRAM, "program"], #program)};`

`stmt: ID ASSIGN^ expr SEMICOLON!;`

`expr: INT (PLUS^ INT)*;`

`}`: Insert code (imaginary token)

`^`: Make node a root node

`!`: Exclude node from AST

`*`: Left recursion (left associativity)

# ANTLR References

<http://www.antlr2.org/doc/trees.html>

<http://www.antlr2.org/doc/sor.html>

<http://www.antlr2.org/javadoc/antlr/collections/AST.html>

<http://www.antlr2.org/doc/err.html>