PROJECT #1

ID CROSS REF

WHAT TO DO:

CREATE A LISTING OF EACH ID DECL W/ ALL ITS USES

E.G.,

```c
{ INT a;
  BOOL b;
  IF (b)
    a = a+1;
}
```

PRODUCES

1: a(INT): 4(2)
2: b(BOOL): 3
ID CROSS REF IS USED FOR REFACTORING AND DEBUGGING

WHAT YOU LEARN:

• REFRESH YOUR JAVA CODING

• LEARN COMPILER CONCEPTS LIKE
  SCANNER
  PARSER
  ABSTRACT SYNTAX TREE (AST)
  BLOCK STRUCTURED SYMBOL TABLE
WHAT IS CSX-LITE?

SMALL SUBSET OF CSX

STRUCTURE:

PROGRAM IS:

\{ DECLS STMTS \}

A DECL IS:

INT id; OR

BOOL id;

An id IS:

LETTER FOLLOWED BY
LETTERS & DIGITS

E.g. a Kyz G0TEAM CVN76

CASE IS IGNORED!
STATEMENTS ARE

(1) ASSIGNMENT:

\[ \text{id} = \text{EXPR} ; \]

(2) CONDITIONAL:

\[ \text{IF (EXPR) STMT ;} \]

(3) BLOCK:

\[ \{ \text{DECLS STMTS} \} \]

EXPRS ARE:

(1) \text{id}

(2) \text{INT LITERAL}

\[ 1 \quad 0 \quad 777 \]

(3) BINARY EXPRES:

\[ \text{EXPR OP EXPR} \]

\[ a + b \quad a - (b - c) \]

(4) OPS ARE

\[ + \quad - \quad == \quad != \]
PARENS ARE ALLOWED

(a+b) (a+(b+c))

COMMENTS ARE SINGLE LINE:

// TEXT OF COMMENT
WE GIVE YOU A WORKING CSX-LITE SCANNER THAT BUILDS A VALID AST.

YOU NEED NOT HANDLE INVALID PROGRAMS!

YOU WALK THE AST TO GATHER DECL & USE INFO
```c
{ int a;
  { int b;
    b = a;
  }
}
```
BLOCK STRUCTURED
SYMBOL TABLES

Scopes Nest:

\[
\begin{align*}
\text{\{ int } a \\
\quad \text{\{ bool } b \\
\quad \text{if } (b) \ a = 1; // \text{ Here} \\
\quad \text{\}} \\
\text{\}}
\end{align*}
\]

Lookup is innermost to outermost. At point "here", try inner scope, then outer one.
EACH SCOPE HAS A
SYMBOL TABLE. IT MAPS ID NAME
TO ID INFO.

SYMBOL TABLES ARE LINKED
TO REFLECT NESTING.

INNER

\[ B \rightarrow \text{BOOL} \]

OUTER

\[ a \rightarrow \text{INT} \]
TO GET STARTED WE GIVE YOU A WORKING SOLUTION TO A SIMPLER PROBLEM:

COUNT DECLS AND USES PER SCOPE (w/o scoping rules!)

{ INTO; INT4;
  ...
  { INTO;
    c = b + c; 33

SCOPE 1 (LINE 1): 2 DECLS, 0 USES
SCOPE 2 (LINE 3): 1 DECL, 3 USES

↑ IGNORES SCOPING!