Syntax: Static semantics (just syntax that cant be represented by BNF)

Ex: you must declare a variable before you can use it

<procedure body> 🡪 <variable declaration> <rest of body>

<statement-list> 🡪 <statement> | <statement-list>

<statement> 🡪 <declare> | <assign> | …

<declare> 🡪 <type> <name>

<assign> 🡪 <name> = <expression>

<expression> 🡪 <A> 1…

<A> 🡪 <A> + <B> | <A> - <B> | <B>

<B> 🡪 <B> \* <C> | <B> / <C> | <C>

<C> 🡪 <number> | <name>

double 2  
int x  
x = 3\*2+10

<statement list>  
|  
<statement list><statement>  
| \  
<statement list><statement> <assign>  
| | \  
<statement> <declare> <name> = <expression>  
| | | \  
<declare> <type><name> x <A>   
| | | |  
<type><name> int x <A> + <B>  
| | | |  
double 2 <B> <C>  
 / \  
 <B> \* <C> <number>  
 | | |  
 <C> <name> 10  
 / |   
 <number> <Z>  
 |  
 3

Static Semantics: we declare the parse tree by defining the attributes at each of the nodes

An attribute is either  
1. Inherited: an attribute value is determined by parent node and left sibling  
2. Synthesized: an attribute value is determined by children nodes

To verify all vars declared before use we need 2 attributes.

A synthesized attribute to send declaration info up the tree.  
🡪 declaration = the name of the variable being declared  
  
An inherited attribute that collects the declaration info + brings it down tree  
🡪 a declare table = a table of all declared variables