Grammars & Variable Attributes

* When static scope is used, nonlocal variable references are resolved by looking at the syntactic block structure of the program.
* The lifetime of a variable is the time during which the variable is bound to a memory location
* When dynamic scope is used, nonlocal variable references are resolved by looking at which functions called which other functions at run-time.
* The BNF grammar rule A -> B [ C ] means A can be replaced by one B, optionally followed by one C
* The BNF grammar rule A -> B | C means A can be replaced by either one B or one C
* The BNF grammar rule A -> { B C } means A can be replaced by zero or more Bs, followed by zero or more Cs
* The BNF grammar rule A -> B C means A can be replaced by one B, followed by one C
* An example of an explicit heap-dynamic variable is A variable that is allocated at run-time using a command like *new*
* The BNF grammar rule A -> B { C } means A can be replaced by one B, followed by zero or more Cs
* Two of the strings in the language of this grammar are aab, aac

S -> A B  
A -> a a  
B -> b | c

* BNF grammars and syntax diagrams are used to describe the syntax of programming languages
* A static variable is a variable whose lifetime is the entire program run.
* A languages is case-sensitive if upper-case and lower-case versions of identifiers are different identifiers.
* Two of the strings in the language of this grammar are ccdddcc, ccdcc

S -> A B A  
A -> c c  
B -> d B  
B -> d

* The scope of a variable is the portion of the program in which the variable can be referenced
* A grammar is *ambiguous* if there is some string that can be derived from the grammar with two or more parse trees