Dynamic Scoping

In a language with dynamic scoping, the bindings between names and objects depend on the flow of control at run time, and in particular on the order in which subroutines are called. Dynamic scope rules are generally simple: the ”current” binding for a given name is the one encountered most recently **during execution**, and not yet destroyed by returning from its scope. Because the flow of control cannot in general be predicted in advanced, the bindings between names and objects in a language with dynamic scoping cannot in general be determined by a compiler. Type checking in expressions and argument checking in subroutine calls, for example, must in general be deferred until run time. To accommodate all these checks, languages with dynamic scoping tend to be **interpreted**, rather than compiled.

With dynamic scoping, errors associated with the referencing environment may not be detected until run time.