Scope Rules

Scope: The textual region of the program in which a **binding** is active. In most programming languages, the *scope* of a binding is determined **statically** (at runtime). A scope is the body of a **module, class, subroutine,** or **structured control-flow** statement. A scope is sometimes called a **block**.

referencing environment: The set of active **bindings**. *referencing environment* is principally determined by static or dynamic **scope rules**. A referencing environment generally corresponds to a sequence of scopes that can be examined (in order) to find the current binding for a given name.

In some case, referencing environments depends on **binding rules**. Specifically, when a reference to a subroutine S is stored in a variable, passed as a parameter to another subroutine, or returned as a function value, one needs to determine when the referencing environment for S is chosen— that is, when the binding between the reference to S and the referencing environment of S is made.

Two Principle Binding Rule Options

Deep binding: The choice is made when the reference is first created.

Shallow binding: the choice is made when the **reference** is finally used.