Naming Concepts

Three terms are used to describe the characteristics of a variable or function name:

1. **Scope**: where the name is **visible**.
    a. Variables with **block scope** are visible from the point of their declaration to the end of the block where they are declared. Local variables and parameters have block scope.
    b. Functions and global variables have **file scope**; they are visible from the point of declaration to the end of the file in which they are declared.
2. **Storage** and **duration**: where a variable is located, and how long it stays there.
    a. Variables in **static storage** are placed there when the program starts running and stay at the same address until the program is finished. Global variables and local `static` variables have static storage class.
    b. Variables with **automatic storage**, are placed on the stack when they are defined, and then destroyed when the block they are defined in ends; automatic variables always have block scope. Local variables and parameters have automatic storage.
    c. **Dynamic storage** is determined by the programmer; dynamic variables are placed on the heap and removed from the heap in response to specific programmer commands, such as `new` and `delete`.
3. **Linkage**: how variables and functions can be shared between different files.
    a. **External linkage** means that a global variable or function can be used from other files. This is the normal case with global variables.
    b. **Internal linkage** means that a "global" variable or function is only visible to other functions in the same file. This is indicated by placing the keyword `static` before the definition of the variable or function.
    c. **No linkage** means that a variable cannot be used inside any other function. All local variables and parameters have no linkage.