block is executed: storage is deallocated when the block terminates, causing the variable to lose its value. A variable with ***static storage duration*** stays at the same storage location as long as the program is running, allowing it to retain its value indefinitely.

- ***Scope.*** The scope of a variable is the portion of the program text in which the variable can be referenced. A variable can have either ***block scope*** (the variable is visible from its point of declaration to the end of the enclosing block) or ***file scope*** (the variable is visible from its point of declaration to the end of the enclosing file).
- ***Linkage.*** The linkage of a variable determines the extent to which it can be shared by different parts of a program. A variable with ***external linkage*** may be shared by several (perhaps all) files in a program. A variable with ***internal linkage*** is restricted to a single file, but may be shared by the functions in that file. (If a variable with the same name appears in another file, it's treated as a different variable.) A variable with ***no linkage*** belongs to a single function and can't be shared at all.

The default storage duration, scope, and linkage of a variable depend on where it's declared:

- Variables declared *inside* a block (including a function body) have *automatic* storage duration, *block* scope, and *no* linkage.
- Variables declared *outside* any block, at the outermost level of a program, have *static* storage duration, *file* scope, and *external* linkage.

The following example shows the default properties of the variables `i` and `j`:

```
int i;    /* static storage duration, file scope, external linkage */

void f(void)
{
  int j;  /* automatic storage duration, block scope, no linkage */
}
```

For many variables, the default storage duration, scope, and linkage are satisfactory. When they aren't, we can alter these properties by specifying an explicit storage class: `auto`, `static`, `extern`, or `register`.

The `auto` Storage Class

The `auto` storage class is legal only for variables that belong to a block. An `auto` variable has automatic storage duration (not surprisingly), block scope, and no linkage. The `auto` storage class is almost never specified explicitly, since it's the default for variables declared inside a block.