# Princeton University
# COS 217: Introduction to Programming Systems
# C Variable Declarations and Definitions

Variable **declaration**: A statement that informs the compiler of the name, type, scope, linkage, and duration of the variable.
Variable **definition**: A declaration that causes the compiler to allocate storage.

**Scope** (compiletime concept):

**file**: The variable is accessible within the file in which it is declared, from the point of declaration to the end of the file.
**block**: The variable is accessible within the block in which it is declared, from the point of declaration to the end of the block.

**Linkage** (linktime concept):

**external**: The variable is known to the linker, and so is accessible from multiple files.
**internal**: The variable is not known to the linker, and so is accessible from only the file in which it is declared.

**Duration** (runtime concept):

**temporary**: The variable exists only during the execution of the function or block in which it is declared (i.e., the variable is stored in the runtime Stack).
**process**: The variable exists throughout the entire process (i.e., the variable is stored in the Data or BSS Sections). The variable is initialized at program startup; the default initial value is 0.

| C Code | Decl/Def | Scope | Linkage | Duration | Comment |
|---|---|---|---|---|---|
| `int a;` | definition | file | external | process | |
| `extern int b;` | declaration | file | external | process | |
| `static int c;` | definition | file | internal | process | |
| `void fun(int d)` | definition | block | internal | temporary | Common |
| `{` | | | | | |
| `   int e;` | definition | block | internal | temporary | Common |
| `   auto int f;` | definition | block | internal | temporary | Same as default |
| `   extern int g;` | declaration | block | (unknown) | process | Very rare |
| `   static int h;` | definition | block | internal | process | |
| `   ...` | | | | | |
| `}` | | | | | |