

Declaration Qualifiers

Storage Class

Last updated 7/25/25

Declaration Qualifiers – Storage Class

- Declaration – Full Structure
 - This applies to variables and functions
 - Declarations consist of 4 elements
 - Storage Class
 - Type Qualifier
 - Type Specifier
 - Declarator
 - Format
 - (Storage_Class) (Type_Qualifier) Type_Specifier Declarator;
int foo;
 - () indicates optional

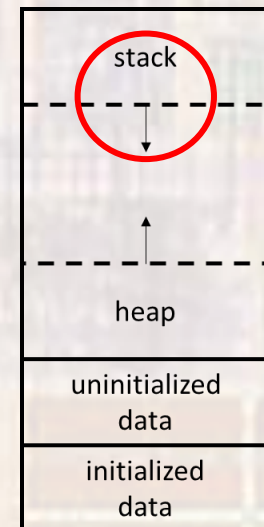
Declaration Qualifiers – Storage Class

- Storage Class
 - The storage class determines the duration of a variable's existence – it's scope
 - Storage class dictates where the variable is stored in data memory
 - Storage class determines where the variable can be seen
 - no linkage – only visible in the current block {}
 - internal linkage – visible in the current file
 - external linkage – visible in more than one file
 - 4 storage classes
 - Auto
 - Static
 - Extern
 - Register

Declaration Qualifiers – Storage Class

- Storage Class - Auto
 - Default value if not specified
 - Variable exists until the end of the block {} (scope) in which it was declared
 - Once the block is completed – the variable no longer exists
 - Only visible in the current file (no linkage)
 - Stored in the **Stack**
 - Auto is typically omitted since it is the default

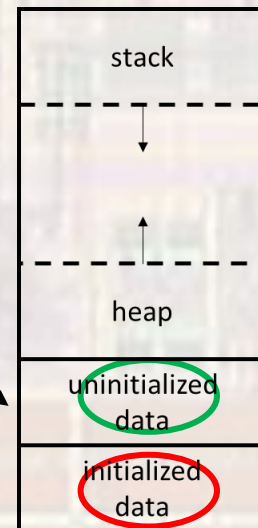
`auto int foo;` \leftrightarrow `int foo;`



Declaration Qualifiers – Storage Class

- Storage Class - Static
 - Variable exists in the block {} in which it was declared for the duration of the program
 - Commonly used in functions to retain the variable's value between calls (e.g. a counter) – (no linkage)
 - Global variables are static
 - Only visible in the current file (internal linkage)
 - Can be initialized at the same time it is declared
 - One of two exceptions to our prescribed declaration format
 - If un-initialized or initialized to 0 – stored in the **BSS** segment
 - If initialized (not 0) – stored in the **Data** segment

```
static int foo = 5;  
static int boo;
```



Declaration Qualifiers – Storage Class

- Storage Class - Extern
 - Variable visible in more than one file (external linkage)
 - Used when multiple files need access to a single variable
 - Defined in 1 file, declared as extern in any files that want to access it
 - Has a **Static** duration
 - Stored in the **Data** segment
 - If it needs to be initialized, it should be done at the same time it is defined – ensures all files see the initialization

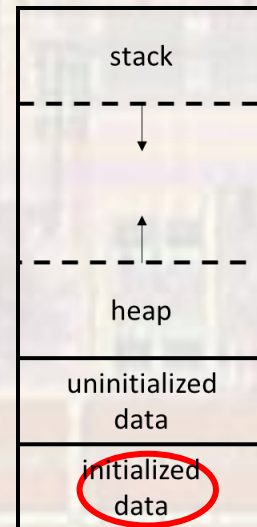
File 1

```
int foo;  
int loo = 3;  
int boo[10]
```

File 2,3,...

```
extern int foo;  
extern int loo;  
extern int boo[]
```

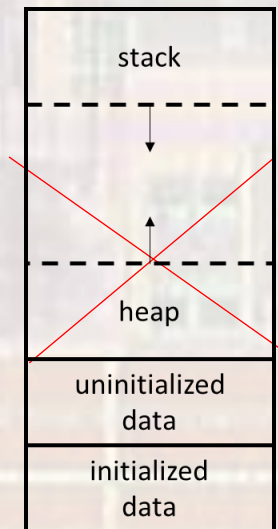
note – no size here



Declaration Qualifiers – Storage Class

- Storage Class - Register
 - Variable stored in a CPU register
 - Allows fast access
 - Only defined with-in a block {}
 - Only visible in the current block (no linkage)
 - Not used often due to high quality compilers used today
 - The compiler can ignore this request

register int foo;



Declaration Qualifiers – Storage Class

- Data Memory

