ESP_LEC_2

Local Variable: The variables declared inside a block are automatic or local variables. The local variables exist only inside the block in which it is declared.

Global Variable: Variables that are declared outside of all functions are known as external or global variables. They are accessible from any function inside the program.

Storage class: Storage class in C determines the scope of a variable. The storage class of a variable in C determines the life time of the variable

There are four storage class specifiers in C

Auto/Automatic Storage Class

- A variable defined within a function or block with auto specifier belongs to automatic storage class. All variables defined within a function or block by default belong to automatic storage class if no storage class is mentioned. Variables having automatic storage class are local to the block which they are defined in, and get destroyed on exit from the block.

Register

- The register specifier declares a variable of register storage class. Variables belonging to register storage class are local to the block which they are defined in, and get destroyed on exit from the block. A register declaration is equivalent to an auto declaration, but hints that the declared variable will be accessed frequently; therefore they are placed in CPU registers, not in memory.
- Only a few variables are actually placed into registers, and only certain types are eligible.
- If a variable is declared register, the unary & (address of) operator may not be applied to it, explicitly or implicitly. Register variables are also given no initial value by the compiler.

Static

- The static specifier gives the declared variable static storage class. Static variables can be used within function or file. Unlike global variables, static variables are not visible outside their function or file, but they maintain their values between calls. The static specifier has different effects upon local and global variables.

- When static specifier is applied to a local variable inside a function or block, the compiler creates permanent storage for it, much as it creates storage for a global variable but static local variable remains visible only to the function or block in which it is defined. In simple terms, a static local variable is a local variable that retains its value between function calls.
- When static specifier is applied to a global variable or a function then compiler makes that variable or function known only to the file in which it is defined.
- Static variables have default initial value zero and initialized only once in their lifetime.

Extern

- The extern specifier gives the declared variable external storage class. The principal use of extern is to specify that a variable is declared with *external linkage* elsewhere in the program.
- When extern specifier is used with a variable declaration then no storage is allocated to that variable and it is assumed that the variable has already been defined elsewhere in the program. When we use extern specifier the variable cannot be initialized because with extern specifier variable is declared, not defined.
- extern can also be applied to a function declaration, but doing so is redundant because all function declarations are implicitly extern.

Additional -the difference between a declaration and a definition. A declaration declares the name and type of a variable or function. A definition causes storage to be allocated for the variable or the body of the function to be defined. The same variable or function may have many declarations, but there can be only one definition for that variable or function.

Storage Class	Declaration	Storage	Default Initial Value	Scope	Lifetime
auto	Inside a function/block	Memory	Unpredictable	Within the function/block	Within the function/block
register	Inside a function/block	CPU Registers	Garbage	Within the function/block	Within the function/block
extern	Outside all functions	Memory	Zero	Entire the file and other files where the variable is declared as extern	program runtime
Static (local)	Inside a function/block	Memory	Zero	Within the function/block	program runtime
Static (global)	Outside all functions	Memory	Zero	Within the file	program runtime