Functions - Local and Global Variables

Local Variables

- Variables defined inside a function are known as automatic local variables.
  - ➔ They are automatically “created” each time the function is called .
  - ➔ Their values are local to the function.
- The Values of a local variable can only be accessed by the function in which the defined.
  - ➔ Its value cannot be accessed by any other function.
- If an initial value is given to a variable inside a function, that initial value is assigned to the variable each time the function is called .
- Can use the auto keyword to be more precise, but not necessary, as the compiler adds this by default.
- Local variables are also applicable to any code where the variable is created in a block (loops, if statements)

Global Variables:

- The opposite of a variable
- Global variables value can be accessed by any function in the program.
- A Global variable are declared outside of any function,
  - ➔ Does not belong to any particular function.
- Any function in the program can change the value of a global variable .
- If there is a local variable declared in a function with the same name, then , within that function the local variable will mask the global variable.
  - ➔ Global variable is not accessible and prevent it from being accessed normally.

Example

```
int myglobal = 0;       // global variable

int main ()
{
   int myLocalMain = 0;       // local variable
  // can access my global and myLocal
  return 0;
}

void myFunction()
{
  int x;         // local variable
 // can access myGlobal and x, cannot access myLocal
}
```

- In general, Global variables are a “bad” thing and should be avoided .
  - ★ Promotes coupling between functions. (dependencies)

    - ★ Hard to find the location of a bug on a program
    - ★ Hard to fix a bug once its found
- Use parameters in functions instead
    - ★ If a lot of data , use a struct.