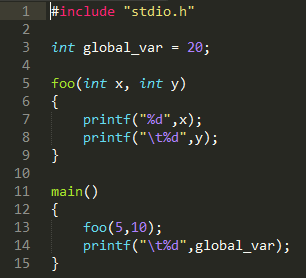
**Local Variables**

Local variables are variables that are declared within a function or block, and are bounded by the scope of that block. Local variables cannot be used, nor accessed, by any code outside the scope it is declared. The variable is created at the start of the function's execution, and destroyed as soon as the program exits the function. This contrasts to global variables. Global variables are accessible anywhere throughout the program, meaning it is not bounded by any particular scope.

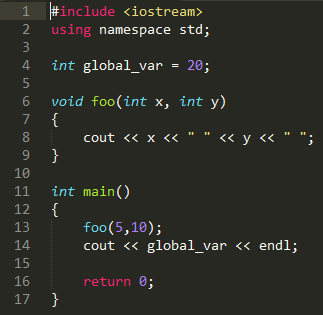
In the following examples we can see the result of using local variables compared to global variables in various programming languages.

* First, we declare a global variable, global\_var.
* Then we define a function, foo, which takes in two arguments, x and y, and prints the given arguments. These arguments are local variables, and can only be used inside this function. If we were to try to print x and y anywhere else outside this function, we will receive an error during compilation. Once again, this is because the variables do not exist outside the function -- they are created and destroyed during the execution of the function only, and will not otherwise be stored in memory.
* Finally, we define the main() function, which calls foo, and passes the values 5 and 10 for the variables x and y respectively. The main() function also prints global\_var. This is allowed, and will not give us an error because global\_var is a global variable and can be used in and out of any function (hence, it was declared outside of any scope).

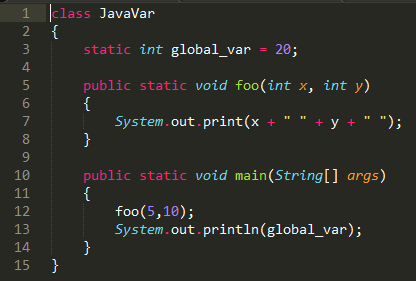
**In C:**

****

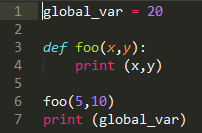
**In C++:**

****

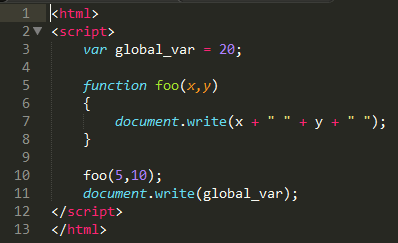
**In Java:**

****

**In Python:**

****

**In JavaScript:**

****

The output of all of these examples are the same. When they are executed, the following numbers are printed (x, y, global\_var),



**Conclusion**

We now see the use of local variables in many examples through languages such as C, C++, Java, Python, and JavaScript. We have shown the importance of scope, and how local variables are used compared to global variables. We have also seen how local variables are stored in memory, when these variables are allocated and de-allocated, and when these variables are accessible.