



Scope of Variables





Scope of Variables

The scope of a variable is the area of code in which the variable is usable

i.e. how much of the program has access to it.

The scope of a variables can be either:

- **Global**
- **Local**

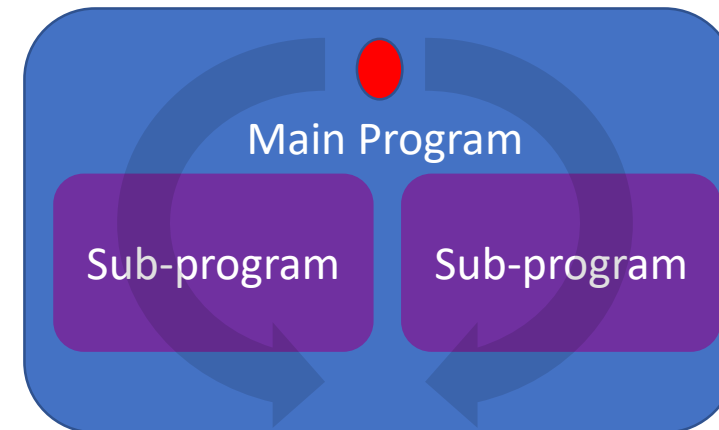


Global Variables

A global variable exists and can be accessed and changed from **any part of the program**.

Global variables do not have to be passed into procedures as parameters because the procedure can access it without doing so.

Global variables **reduce modularity** of a program and should be avoided wherever possible.





Global Variables

The use of global variables reduces modularity because:

- Different programmers could use **conflicting variable names** which would cause errors
- Any procedure could **accidentally alter** a global variable as it doesn't have to be passed in to be used.



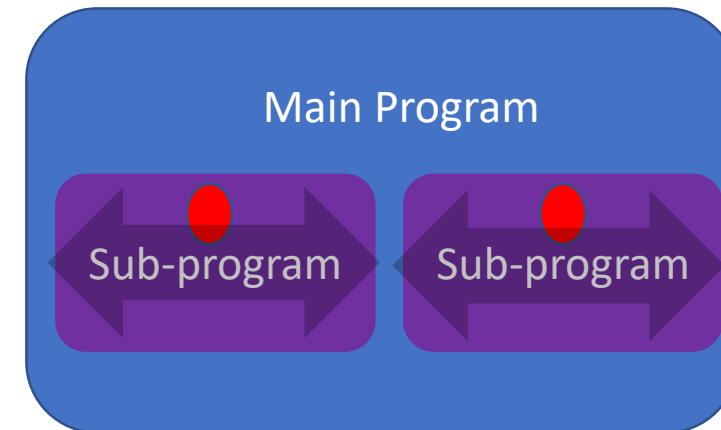
Local Variables

Local variables exist only **within a procedure or function**.

Local variables are declared within a sub-program

They are **not passed in or out** and can only be used within the sub-program they were declared in.

Local variables **cannot be accessed** from out with their own sub-program which limits their scope.





Limiting Scope

It is always preferable to limit the scope of a variable to an individual sub-program wherever possible.

Limiting the scope of a variable is done by:

- Using **local variables** which can only be accessed with their own sub-program
- Using **parameter passing** to only pass to a sub-program the variables it requires.