**Recursivity.**

Recursivity is the property that functions have to be called by themselves. It is useful for some tasks, such as sorting elements, or calculating the factorial of numbers. For example, in order to obtain the factorial of a number (n!) the mathematical formula would be:

*n! = n \* (n-1) \* (n-2) \* (n-3) ... \* 1*

More concretely, 5! (factorial of 5) would be:

*5! = 5 \* 4 \* 3 \* 2 \* 1 = 120*

And a recursive function to calculate this in C++ could be:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 | // factorial calculator  #include <iostream>  using namespace std;  long factorial (long a)  {  if (a > 1)  return (a \* factorial (a-1));  else  return 1;  }  int main ()  {  long number = 9;  cout << number << "! = " << factorial (number);  return 0;  } | 9! = 362880 | [Edit & Run](https://cplusplus.com/doc/tutorial/functions/) |

Notice how in function factorial we included a call to itself, but only if the argument passed was greater than 1, since, otherwise, the function would perform an infinite recursive loop, in which once it arrived to 0, it would continue multiplying by all the negative numbers (probably provoking a stack overflow at some point during runtime).