Recursive Functions

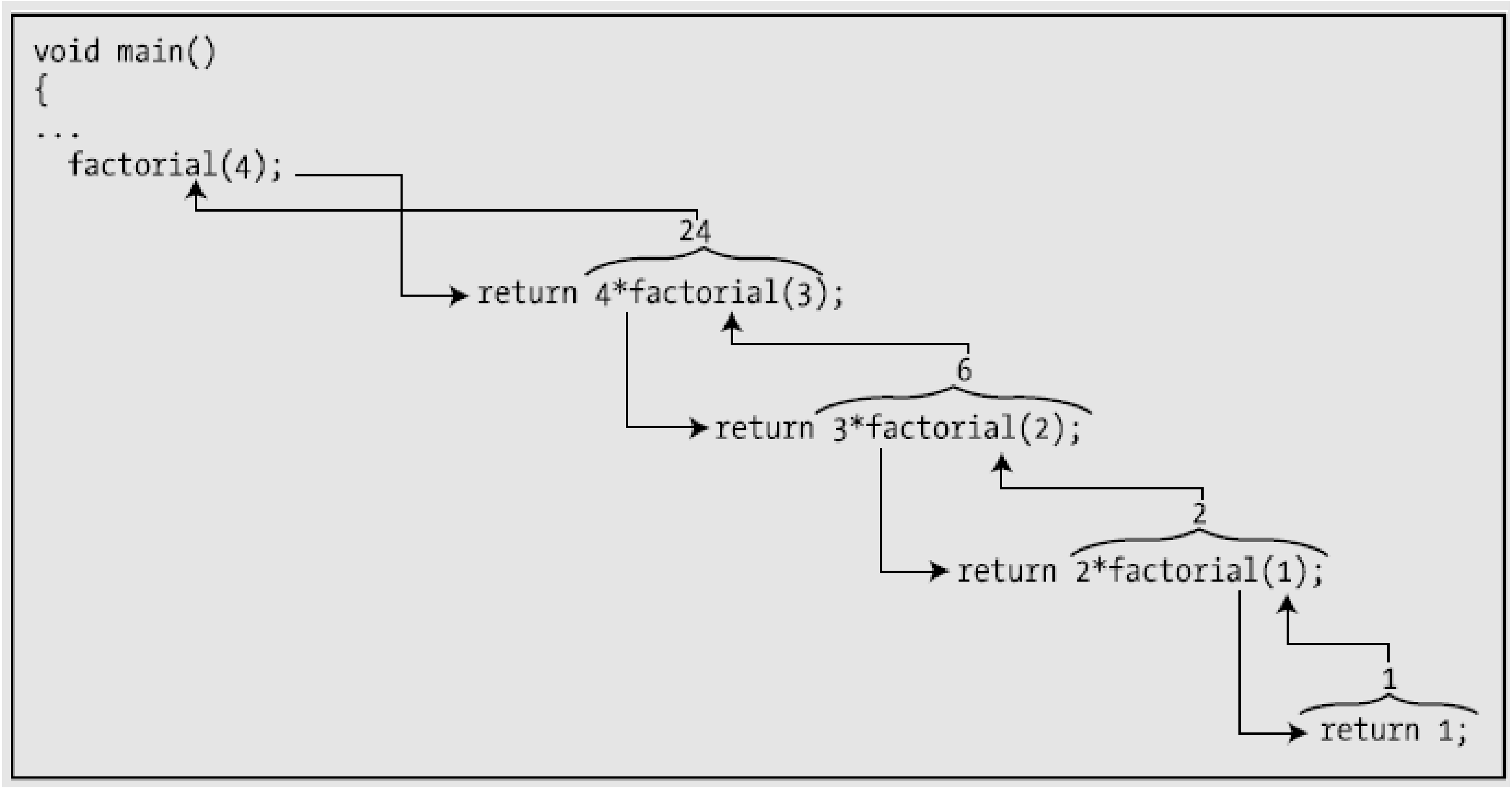
A function can be called not only from by other function by also by itself. Such functions are called **recursive** . A recursive function must have mandatory some condition to stop the recursion, otherwise it will call itself till stack overflow occurs.

Recursive Functions

Example:

```
unsigned int Factorial (unsigned int a)
{
  if (a<2) return a;
  return a*Factorial(a-1);
}
```

Recursive Functions



```
void main()
{
...
   factorial(4);
```

Recursive Functions

- The function factorial() gets called from main() with number having the value 4 as the argument.
- Within the factorial() function itself, because the argument is greater than 1, the statement executed is: <u>return a*Factorial(a-1);</u>
- This is the second return statement in the function, and it calls factorial() again with the argument value 3 from within the arithmetic expression. This expression can't be evaluated, and the return can't be completed until the value is returned from this call to the function factorial() with the argument 3.
- This continues until the argument in the last call of the factorial() function is 1. In this case, the first return statement <u>return a;</u> is executed and the value 1 is returned to the previous call point. This call point is, in fact, inside the second return in the factorial() function, which can now calculate 2 * 1 and return to the previous call.
- In this way, the whole process unwinds, ending up with the value required being returned to main().