**Jump statements.**

Jump statements allow altering the flow of a program by performing jumps to specific locations.

**The break statement**

*break* leaves a loop, even if the condition for its end is not fulfilled. It can be used to end an infinite loop, or to force it to end before its natural end. For example, let's stop the countdown before its natural end:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 | // break loop example  #include <iostream>  using namespace std;  int main ()  {  for (int n=10; n>0; n--)  {  cout << n << ", ";  if (n==3)  {  cout << "countdown aborted!";  break;  }  }  } | 10, 9, 8, 7, 6, 5, 4, 3, countdown aborted! | [Edit & Run](https://cplusplus.com/doc/tutorial/control/) |

**The continue statement**

The *continue* statement causes the program to skip the rest of the loop in the current iteration, as if the end of the statement block had been reached, causing it to jump to the start of the following iteration. For example, let's skip number 5 in our countdown:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 | // continue loop example  #include <iostream>  using namespace std;  int main ()  {  for (int n=10; n>0; n--) {  if (n==5) continue;  cout << n << ", ";  }  cout << "liftoff!\n";  } | 10, 9, 8, 7, 6, 4, 3, 2, 1, liftoff! | [Edit & Run](https://cplusplus.com/doc/tutorial/control/) |

**The goto statement**

*goto* allows to make an absolute jump to another point in the program. This unconditional jump ignores nesting levels, and does not cause any automatic stack unwinding. Therefore, it is a feature to use with care, and preferably within the same block of statements, especially in the presence of local variables.

The destination point is identified by a *label*, which is then used as an argument for the *goto* *statement*. A *label* is made of a valid identifier followed by a colon (*:*). *goto* is generally deemed a low-level feature, with no particular use cases in modern higher-level programming paradigms generally used with C++. But, just as an example, here is a version of our countdown loop using *goto*:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 | // goto loop example  #include <iostream>  using namespace std;  int main ()  {  int n=10;  mylabel:  cout << n << ", ";  n--;  if (n>0) goto mylabel;  cout << "liftoff!\n";  } | 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, liftoff! | [Edit & Run](https://cplusplus.com/doc/tutorial/control/) |