

iteration-statement:
 while-statement
 do-while-statement
 for-statement

while-statement:
 while (*condition*) *statement*

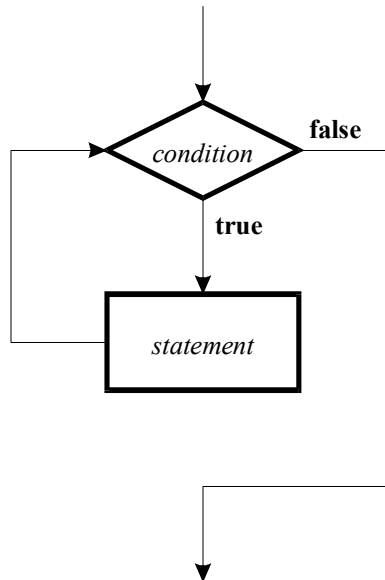


Figure 1. *while-statement* control flow

```
#include <iostream>
using namespace std;
int main()
{
    int count=10;
    while (count > 0) {
        cout << count << ", ";
        count--;
    }
    cout << "Blast Off!";
    cout << endl;
    return 0;
}
```

Figure 2. Program p01, *while-statement* example

Program p01 prints:
10, 9, 8, 7, 6, 5, 4, 3, 2, 1, Blast Off!

do-while-statement:

do *statement* **while** (*expression*) ;

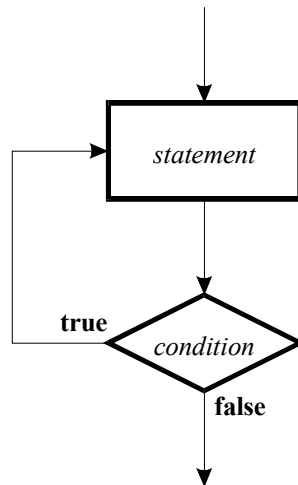


Figure 3. *do-while-statement* control flow.

```
#include <iostream>
using namespace std;
int main()
{
    int count=10;
    do {
        cout << count << ", ";
        count--;
    } while (count > 0);
    cout << "Blast Off!";
    cout << endl;
    return 0;
}
```

Figure 4. Program **p02**: *do-while-statement* example

Program **p02** prints:

10, 9, 8, 7, 6, 5, 4, 3, 2, 1, Blast Off!

for-statement:

for (*initialization-expression-list* ; *condition* ; *increment-decrement-expression-list*) *statement*

initialization-expression-list:

initialization-expression

initialization-expression-list , *initialization-expression*

initialization-expression:

expression-statement

simple-declaration

increment-decrement-expression-list:

expression-statement

increment-decrement-expression-list , *expression-statement*

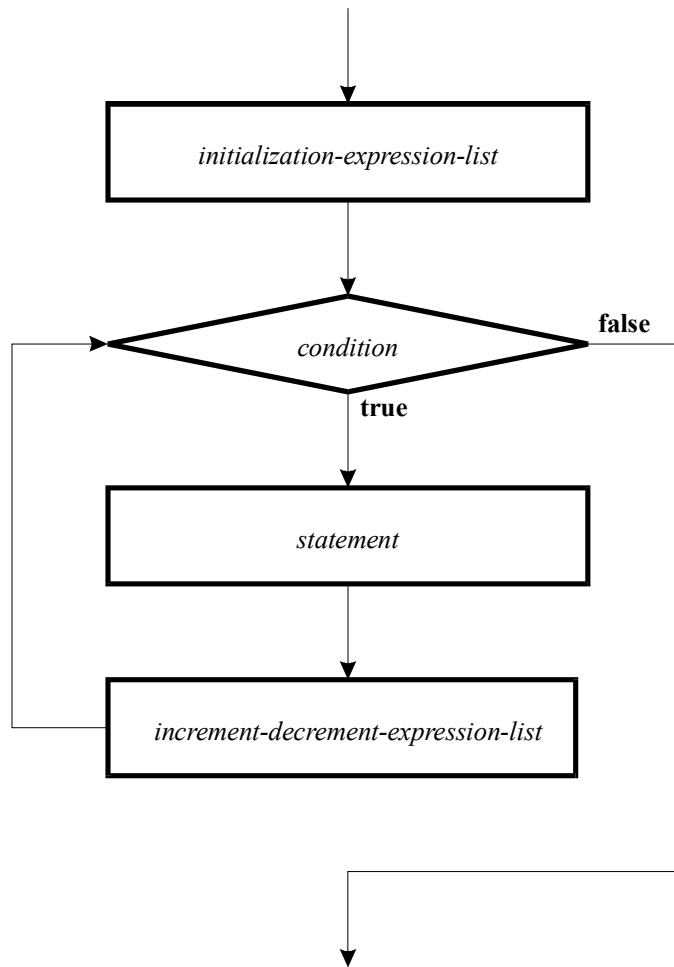


Figure 5. *for-statement* control-flow.

```
#include <iostream>
using namespace std;
int main()
{
    for (int count=10;count>0;count--) cout << count << ", ";
    cout << "Blast Off!";
    cout << endl;
    return 0;
}
```

Figure 6. Program p03: *for-statement* example

Program **p03** prints:

10, 9, 8, 7, 6, 5, 4, 3, 2, 1, Blast Off!