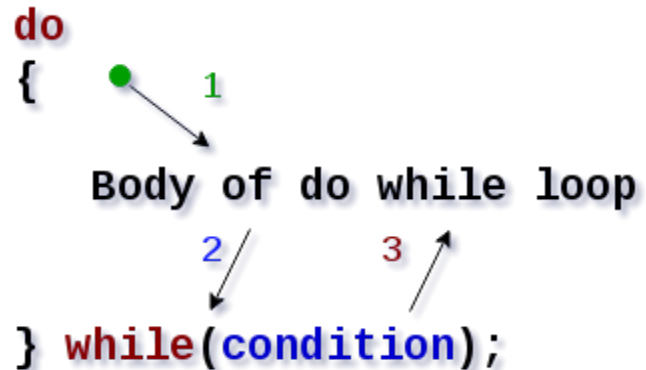


Do While Loop

The **do while** loop is one of the three main forms of looping you will come across. This loop will always execute at least once. Take note that the condition to check whether or not to continue execution of the body of the loop happens at the very end. Another important thing to realize is the **while** condition is outside of the curly brackets. This means that any local variable inside the **do while** loop will not work inside the condition parameters as well as anywhere else in the program. For a general runthrough of a **do while** loop:



Step 1: Execute the body of the do while loop

Step 2: Check condition. True = proceed / False = exit

Step 3: Go to step 1

Remember that a semicolon is placed after the **while** condition.

As an example, this loop can be used when you ask the user for input and want to ask them repeatedly if the wrong response has been given. Here is the code of this example:

```
int input;
do {
    cout << "Enter a number between 1 and 10 inclusive: ";
    cin >> input;
} while(input < 1 || input > 10);
```

As you see, **input** was declared outside of the loop. Again, this is due to the fact that this is a part of our **while** condition and therefore needs to be within the proper scope. Remember, scope is between curly braces. When this portion of code executes, the message displays to the user and awaits a numerical response. If the input was anything other than [1..10], the message will redisplay and await another input.

The difference between the **do while** loop and the **while** loop is the **do while** loop executes once and the **while** loop does not need to execute at all. The above program could be transformed into a **while** loop by setting **input** equal to zero and using the **while** paradigm with the same condition.