A Hasty Loop

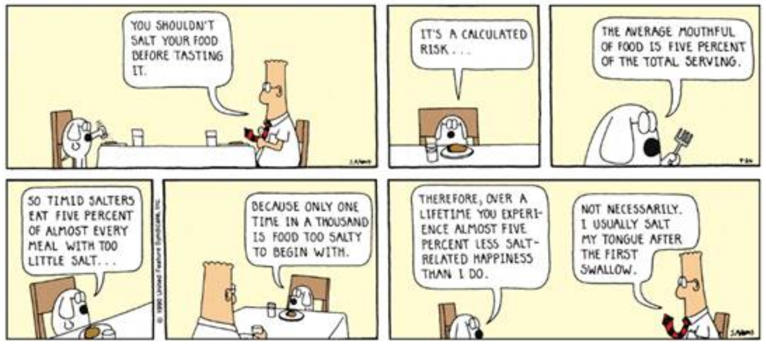
In addition to *for* and *while*, C++ has a loop that tests its condition **after** the loop body completes. The do-while loop always executes the statements inside its body at least once.

```
do
{
    // statements
}
while (condition);
```

The body of the ***do-while*** loop appears between the keywords **do** (which precedes the loop body) and **while**. The body of the ***do-while*** loop can be a single statement, ending with a semicolon, or it can be a compound statement enclosed in braces.

In the ***do-while*** loop, the condition is **followed by a semicolon**, unlike the ***while*** loop, where following the condition with a semicolon leads to subtle, hard tofind bugs.

The ***do-while*** loop is often employed by beginning programmers because it seems more natural. If you find yourself in this situation, think twice. 99% of the time, a ***while*** loop or a ***for*** loop is a better choice than a ***do-while***. In fact, except for salting your food...


YOU SHOULDN'T SALT YOUR FOOD BEFORE TASTING IT.
IT'S A CALCULATED RISK...
THE AVERAGE MOUTHFUL OF FOOD IS FIVE PERCENT OF THE TOTAL SERVING.
SO TIMID SALTERS EAT FIVE PERCENT OF ALMOST EVERY MEAL WITH TOO LITTLE SALT...
BECAUSE ONLY ONE TIME IN A THOUSAND IS FOOD TOO SALTY TO BEGIN WITH.
THEREFORE, OVER A LIFETIME YOU EXPERIENCE ALMOST FIVE PERCENT LESS SALT-RELATED HAPPINESS THAN I DO.
NOT NECESSARILY. I USUALLY SALT MY TONGUE AFTER THE FIRST SWALLOW.

which should **always be done before tasting**, there are relatively few other situations where a test-at-the-bottom strategy is superior to "looking before you leap."