Loops

Today

- String review and getline
- Pre/post increment
- While loops
- do While Loop

Due this week

- **Recitation 3**
- **Homework 3**
  - Write solutions in VSCode and paste in Autograder, **Homework 3 CodeRunner**.
  - Zip your .cpp files and submit on canvas **Homework 3**.
- 3-2-1 on Friday
- Start going through the textbook readings and watch the videos

  - Quiz 4

String Input

Reading two (whitespace separated) strings with cin

```
cout << "Please enter your name: ";
string fname, lname;
cin >> fname >> lname;


//fname gets Harry, lname gets Potter
```

String Input

`getline()` function allows us to accepts a full string input (with whitespace)

```
cout << "Please enter your name: ";
string name;
getline(cin, name);


//name gets Harry Potter
```

String Functions

- The `length` *member function* yields the number of characters in a string.

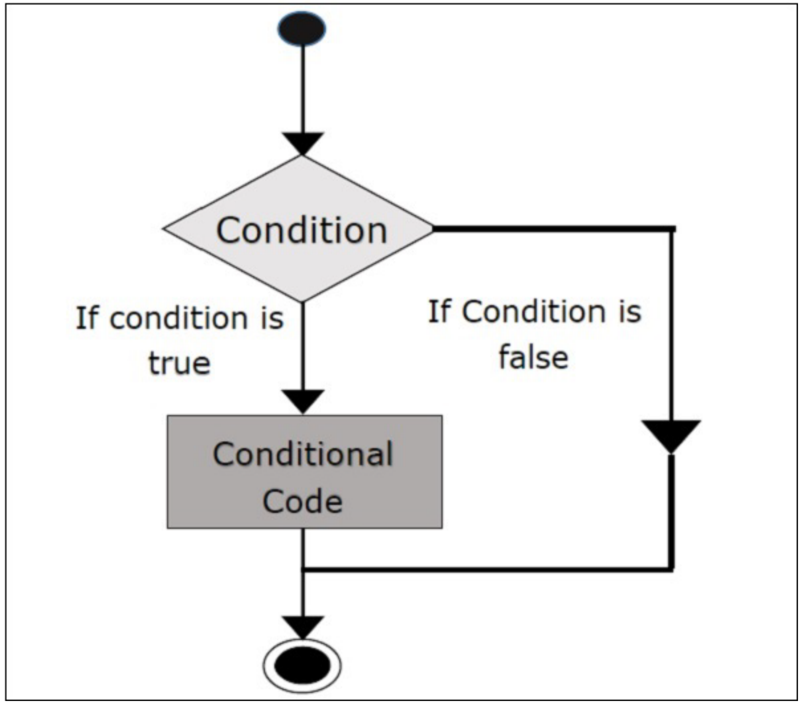
- Unlike the `sqrt` or `pow` function, the `length` function is *invoked* with the *dot notation*:

```
string name = “Harry”;
int n = name.length();
```

String Data Representation & Character Positions

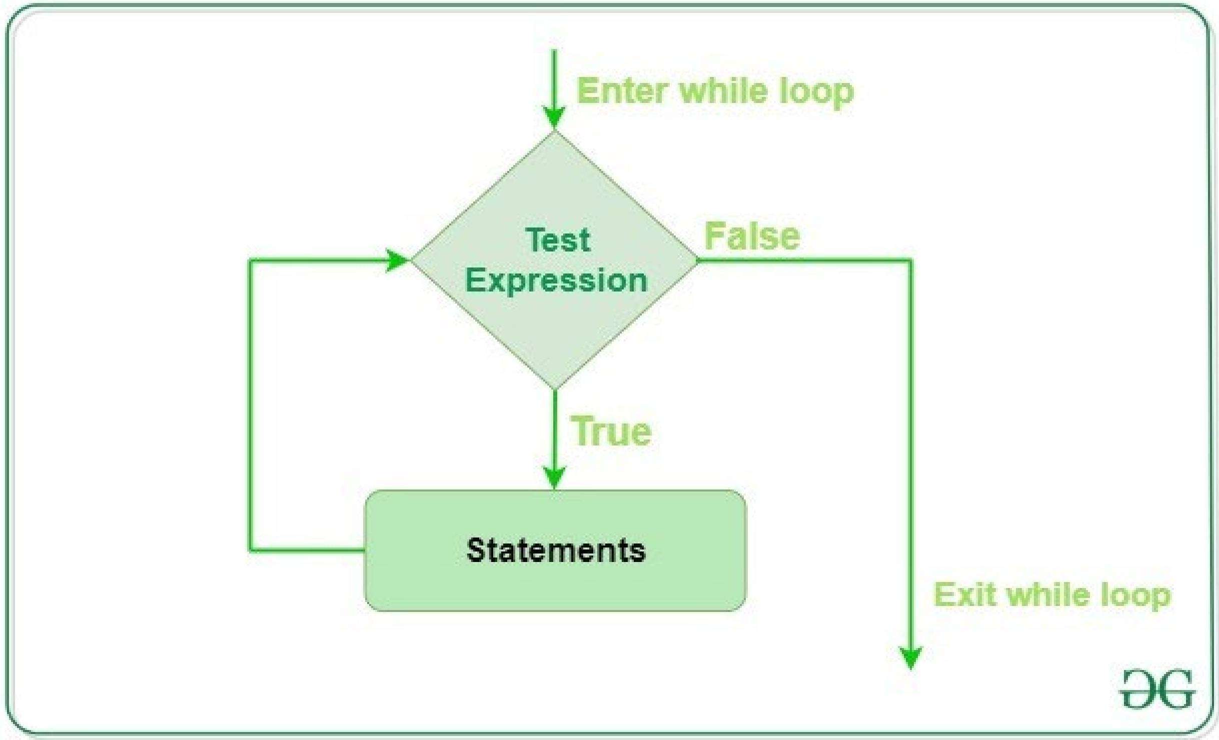
`H`	`e`	`l`	`l`	`o`	`,`		`W`	`o`	`r`	`l`	`d`	`!`
0	1	2	3	4	5	6	7	8	9	10	11	12

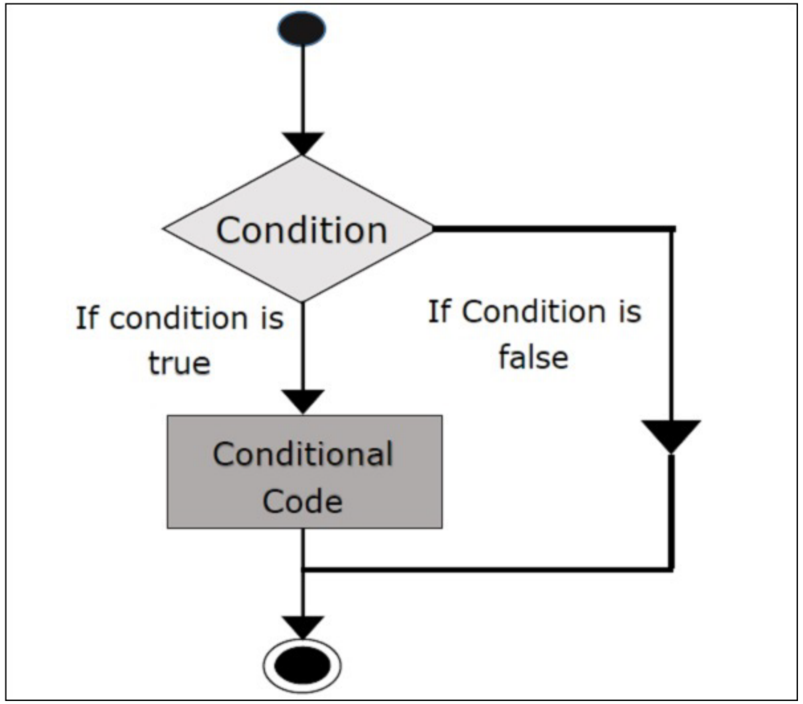
- In most computer languages, the starting position 0 means "start at the beginning."
- The first position in a string is labeled 0, the second 1, and so on. And don't forget to count the space character after the comma—but the quotation marks are **not** stored.
- The position number of the last character is always one less than the length of the **string.**

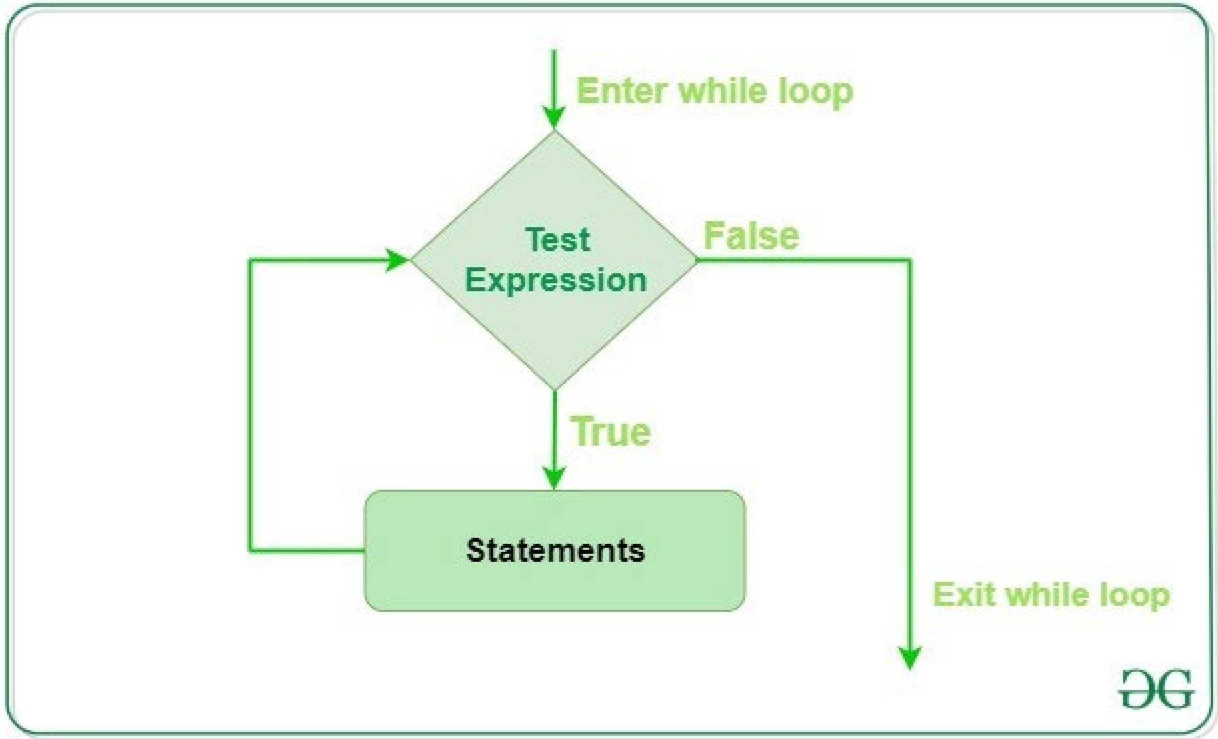


If statement

While loop


Enter while loop
Test Expression
False
True
Statements
Exit while loop

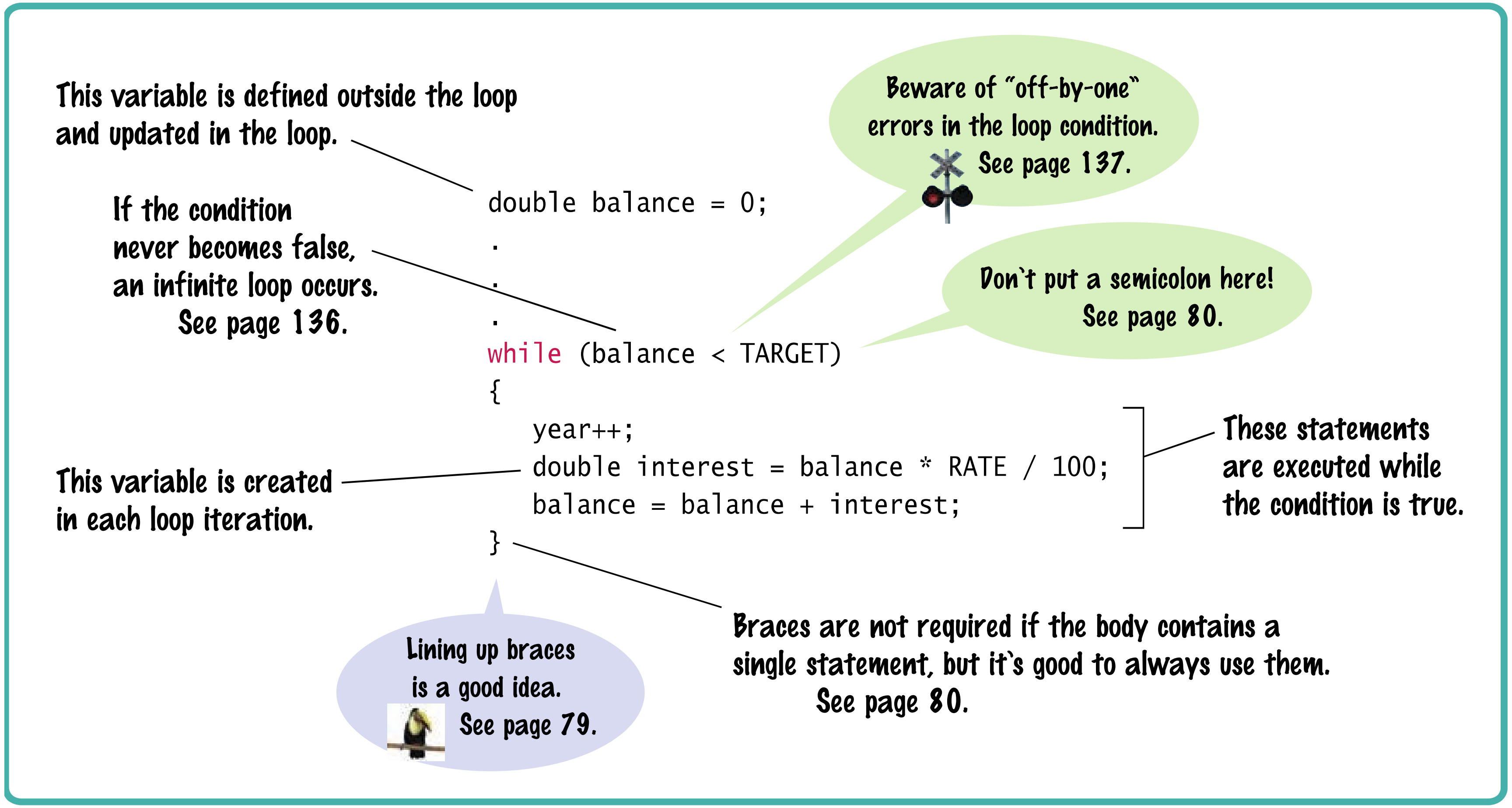

Condition
If condition is true
If Condition is false
Conditional Code


Enter while loop
Test Expression
False
True
Statements
Exit while loop

Where to?

Loops

The *while* Loop Syntax



```
double balance = 0;
.
.
.
while (balance < TARGET)
{
   year++;
   double interest = balance * RATE / 100;
   balance = balance + interest;
}
```

This variable is defined outside the loop and updated in the loop.

If the condition never becomes false, an infinite loop occurs. See page 136.

Beware of "off-by-one" errors in the loop condition. See page 137.

Don't put a semicolon here! See page 80.

These statements are executed while the condition is true.

This variable is created in each loop iteration.

Lining up braces is a good idea. See page 79.

Braces are not required if the body contains a single statement, but it's good to always use them. See page 80.

Brief C++ by Cay Horstmann

while Loop Examples

Loop (all preceded by i=5;)	Output	Explanation

while (i > 0)
{ cout << i << " "; i--; }
	5 4 3 2 1	When i is 0, the loop condition is false, and the loop ends.

while (i > 0)
{ cout << i << " "; i++; }
	5 6 7 8 9 10 11
	...
		The i++ statement is an error causing an “infinite loop” (see Common Error 4.1).

while (i > 5)
{ cout << i << " "; i--; }
	(No output)	The statement i > 5 is false, and the loop is never executed.

while (i < 0)
{ cout << i << " ";
  i--;
}
	(No output)	The programmer probably thought, “Stop when i is less than 0”. However, the loop condition controls when the loop is executed, not when it ends (see Common Error 4.2).

while (i > 0);
{ cout << i << " ";
i--;
}
	(No output, program does not terminate)	Note the semicolon before the {. This loop has an empty body. It runs forever, checking whether i > 0 and doing nothing in the body.

Example of Normal Execution

while loop to hand-trace

```
i = 5;
while (i > 0)
{
   cout << i << " ";
   i--;
}
```

What is the output?

```
..
```

Example of a Problem – An Infinite Loop

The output never ends

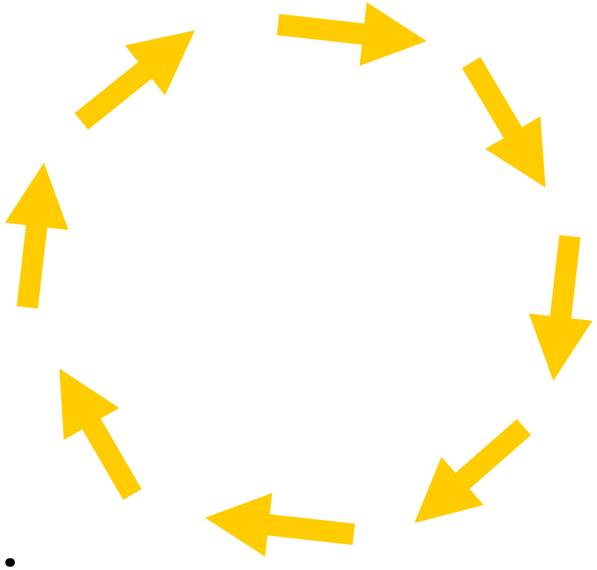
- *i* is set to 5
- The *i++;* statement makes *i* get bigger and bigger
- the condition will never become false –
- an infinite loop

```
i = 5;
while (i > 0)
{
   cout << i << " ";
   i++;
}
```

```
5 6 7 8 9 10 11…
```

Common Error – Infinite Loops

- Forgetting to update the variable used in the condition is common.
- In the investment program, it might look like this:



```
year = 1;
while (year <= 20)
{
   balance = balance * (1 + RATE / 100);
}
```

The variable **year** is not updated in the loop body!

Brief C++ by Cay Horstmann

Another Programmer Error

What is the output?

```
i = 5;
while (i < 0)
{
    cout << i << " ";
    i--;
}
```

A Very Difficult Error to Find (especially after looking for hours and hours!)

What is the output?

```
i = 5;
while (i < 0)
{
     cout << i << " ";
     i--;
}
```

do loop

The `do{ } while()` Loop

- The while() loop’s condition test is the first thing that occurs in its execution.
- The do loop (or do-while loop) has its condition tested only after at least one execution of the statements. The test is at the bottom of the loop:

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

The `do` Loop

- This means that the do loop should be used only when the statements must be executed before there is any knowledge of the condition.

- This also means that the do loop is the least used loop.

do{ } Loop Code: getting user input Repeatedly

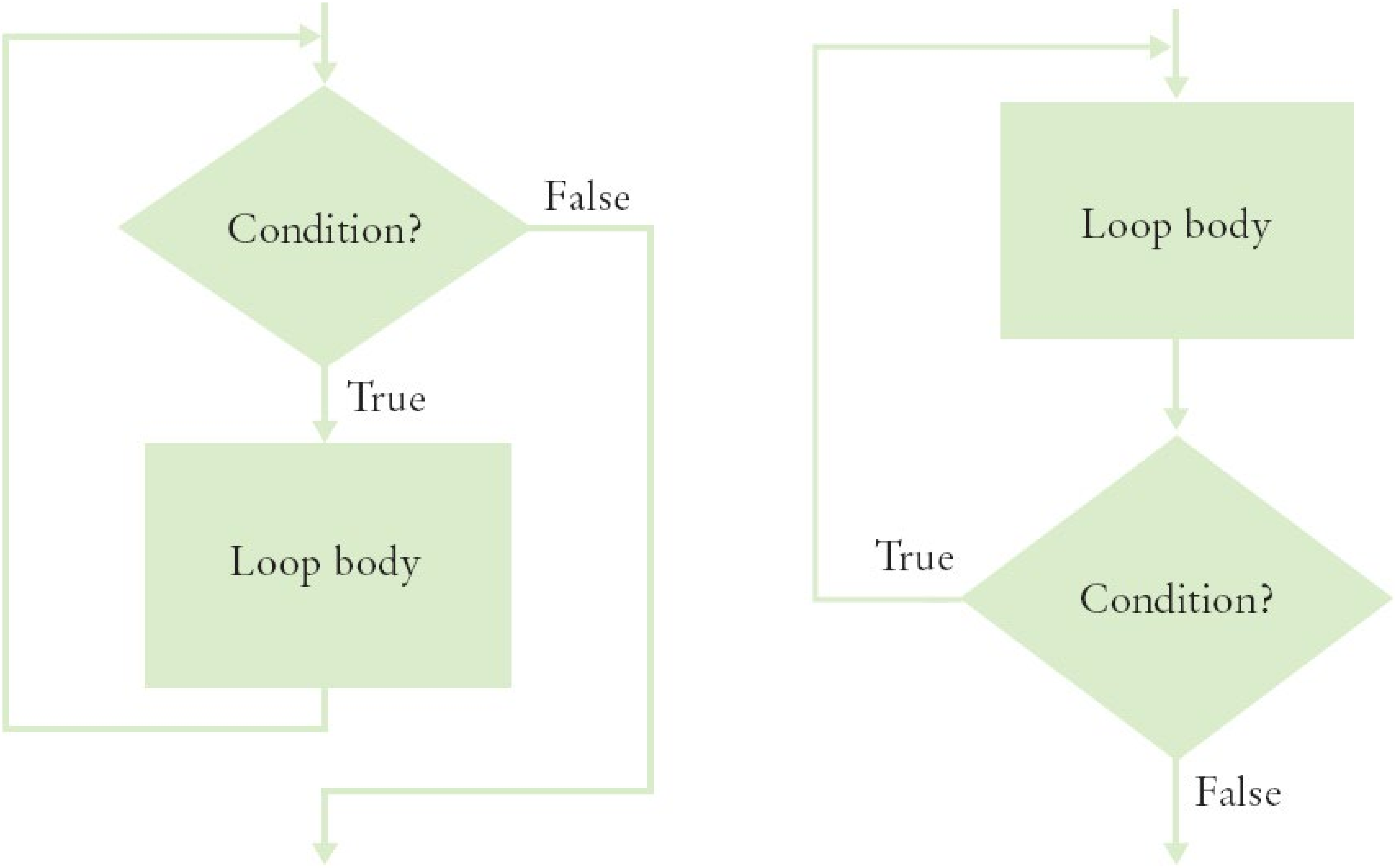
- Code to keep asking a user for input until it satisfies a condition, such as non-negative for applying the sqrt():

```
double value;
do
{
 cout << "Enter a number >= 0: ";
 cin >> value;
}
while (value < 0);

cout << "The square root is " << sqrt(value) << endl;
```

Brief C++ by Cay Horstmann

Flowcharts for the `while` Loop and the `do` Loop



Practice It: Example of do…while

- What output does this loop generate?

```
int j = 1;
do
{
   int value = j * 2;
   j++;
   cout << value << ", ";
 } while (j <= 5);
```

How to Write a Loop

These are the steps to follow when turning a problem description into a code loop:

1. Decide what work must be done inside the loop
   - *For example, read another item or update a total*
2. Specify the loop condition
   - *Such as exhausting a count or invalid input*
3. Determine the loop type
   - *Use for in counting loops, while for event-controlled*
4. Set up variables for entering the loop for the first time
5. Process the result after the loop has finished
6. Trace the loop with typical examples
7. Implement the loop in C++