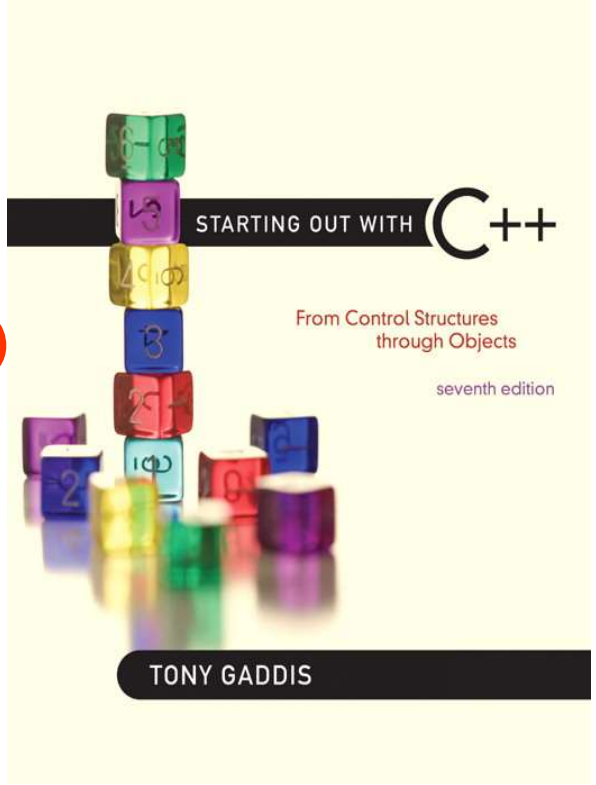
Lecture 13

10 – 10 - 2022



Introduction to Loops: The `while` Loop

Introduction to Loops: The `while` Loop

- Loop: a control structure that causes a statement or statements to repeat
- General format of the `while` loop:

```
while (expression)
    statement;
```

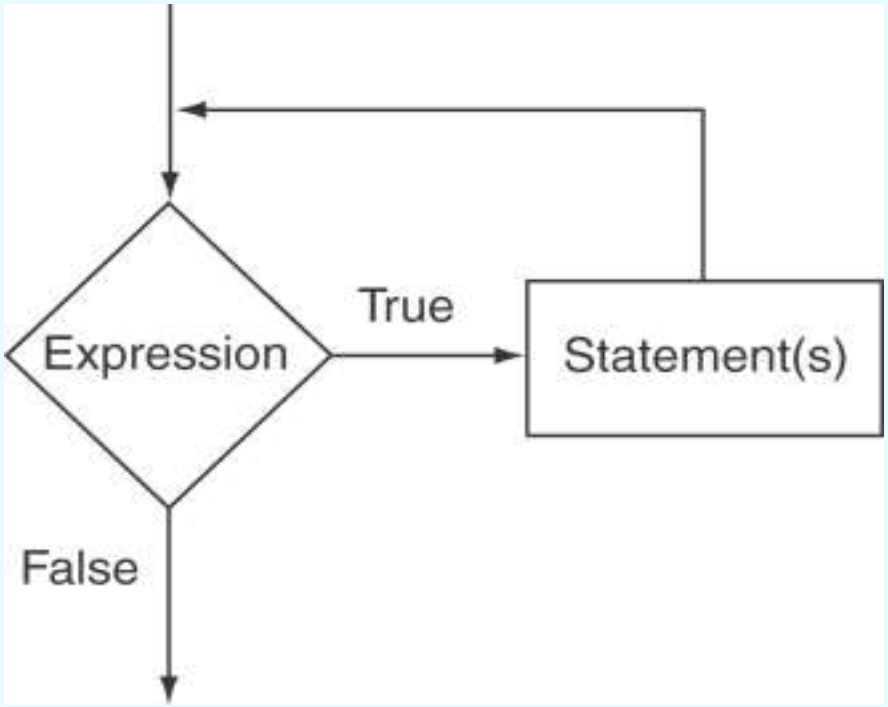
- *`statement;`* can also be a block of statements enclosed in `{ }`

The `while` Loop – How It Works

```
while (expression)
      statement;
```

- *`expression`* is evaluated
  - if `true`, then *`statement`* is executed, and *`expression`* is evaluated again
  - if `false`, then the loop is finished and program statements following *`statement`* execute

The Logic of a `while` Loop



The `while` loop in Program 5-3

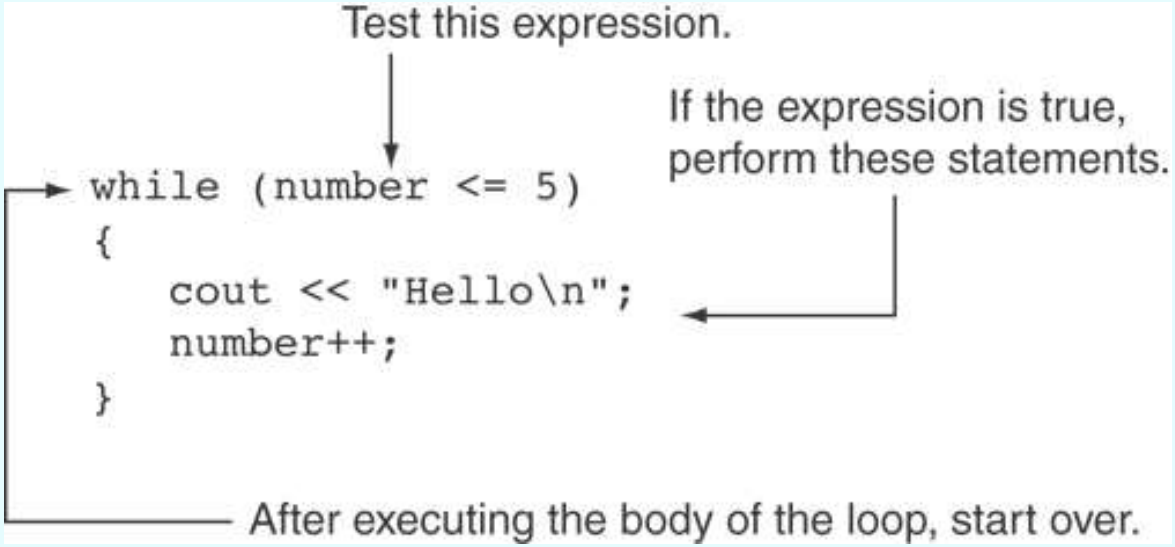
Program 5-3

```
1  // This program demonstrates a simple while loop.
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     int number = 1;
8
9     while (number <= 5)
10    {
11       cout << "Hello\n";
12       number++;
13    }
14    cout << "That's all!\n";
15    return 0;
16 }
```

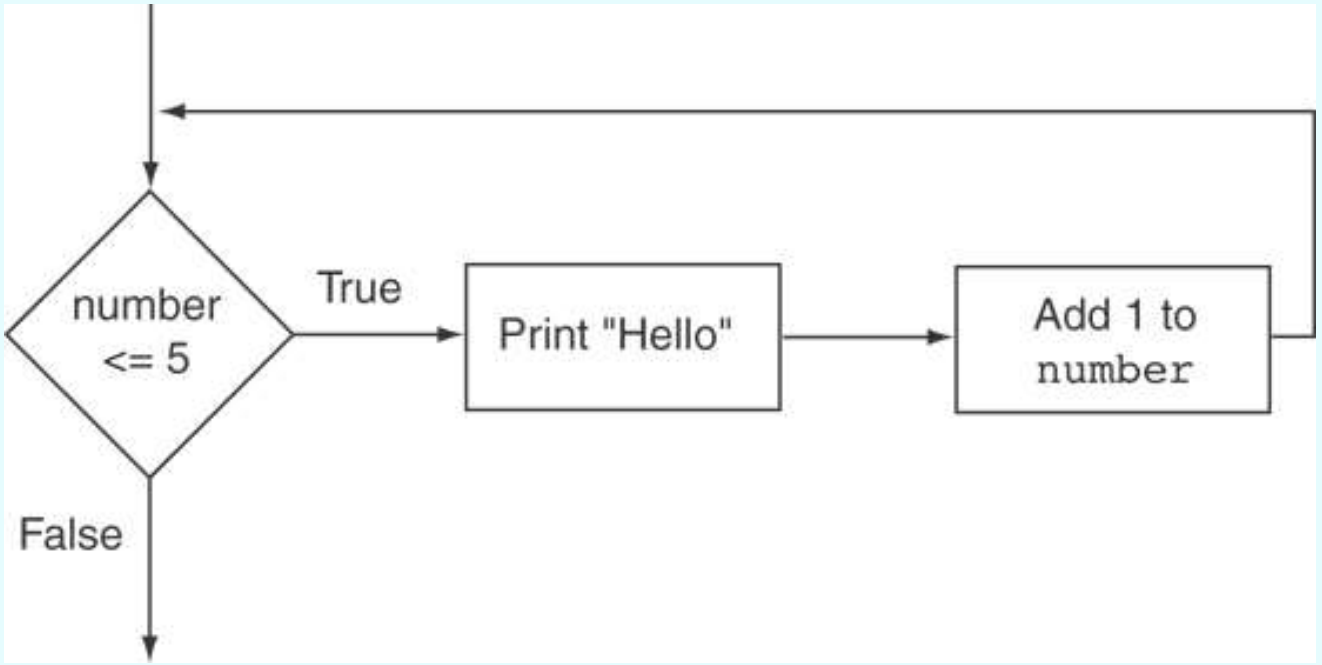
Program Output

```
Hello
Hello
Hello
Hello
Hello
That's all!
```

How the `while` Loop in Program 5-3 Lines 9 through 13 Works



Flowchart of the `while` Loop in Program 5-3



The `while` Loop is a Pretest Loop

`expression` is evaluated *before* the loop executes. The following loop will never execute:

```
int number = 6;
while (number <= 5)
{
   cout << "Hello\n";
   number++;
}
```

Watch Out for Infinite Loops

- The loop must contain code to make *`expression`* become `false`
- Otherwise, the loop will have no way of stopping
- Such a loop is called an *infinite loop*, because it will repeat an infinite number of times

Example of an Infinite Loop

```
int number = 1;
while (number <= 5)
{
   cout << "Hello\n";
}
```

Using the `while` Loop for Input Validation

- Input validation is the process of inspecting data that is given to the program as input and determining whether it is valid.

- The while loop can be used to create input routines that reject invalid data, and repeat until valid data is entered.

Using the `while` Loop for Input Validation

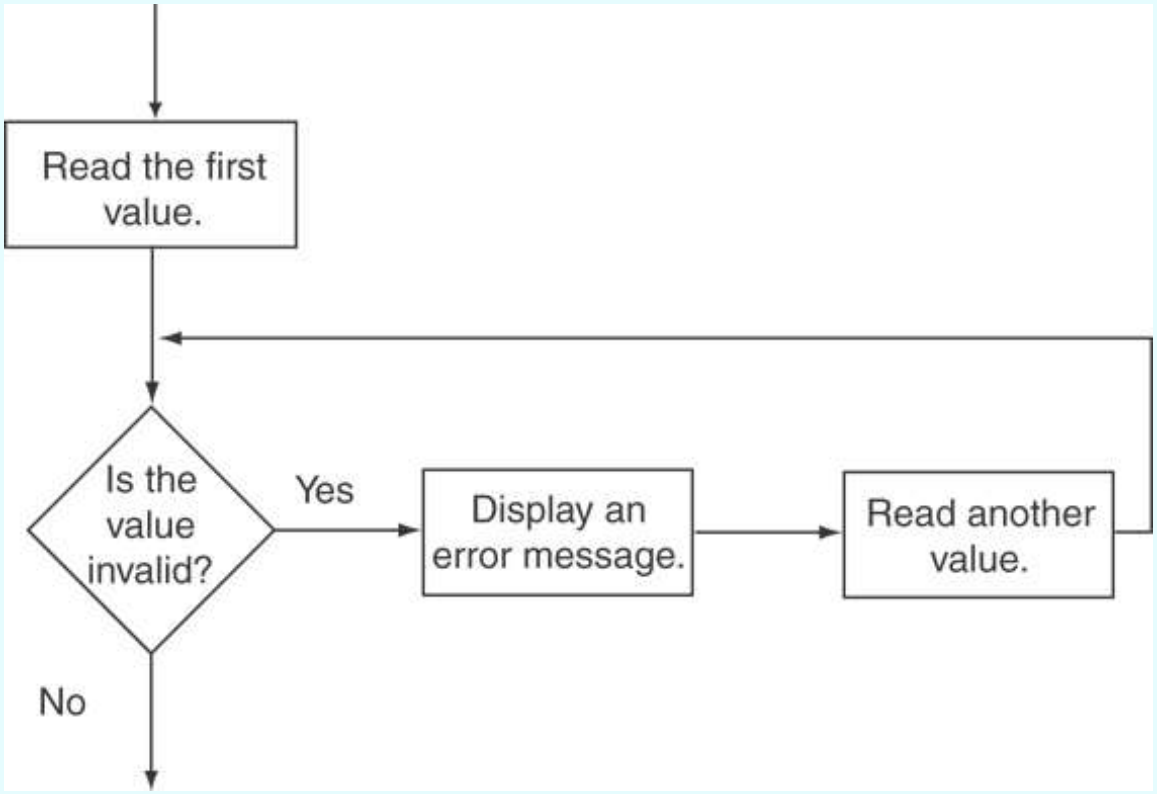
- Here's the general approach, in pseudocode:

Read an item of input.
While the input is invalid
    Display an error message.
    Read the input again.
End While

Input Validation Example

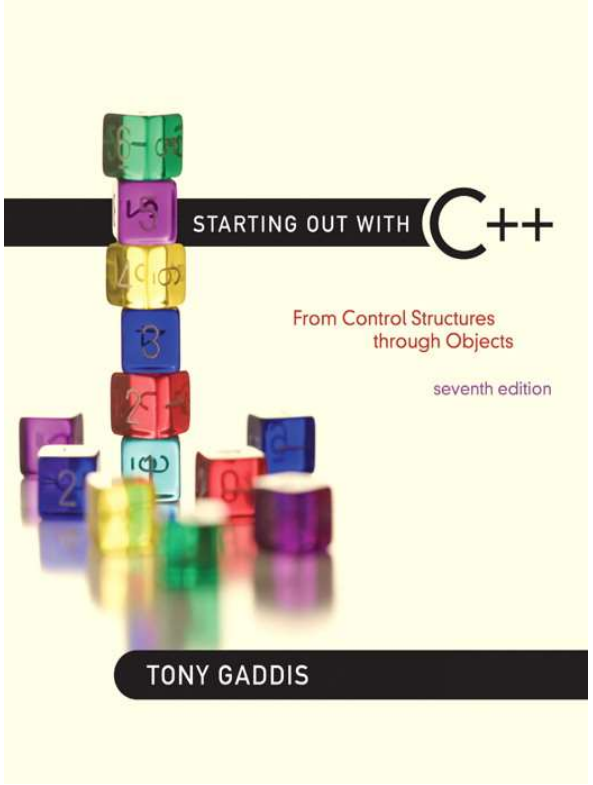
```
cout << "Enter a number less than 10: ";
cin >> number;
while (number >= 10)
{
   cout << "Invalid Entry!"
        << "Enter a number less than 10: ";
   cin >> number;
}
```

Flowchart for Input Validation



Input Validation in Program 5-5

```
20    // Get the number of players per team.
21    cout << "How many players do you wish per team? ";
22    cin >> teamPlayers;
23
24    // Validate the input.
25    while (teamPlayers < MIN_PLAYERS || teamPlayers > MAX_PLAYERS)
26    {
27       // Explain the error.
28       cout << "You should have at least " << MIN_PLAYERS
29            << " but no more than " << MAX_PLAYERS << " per team.\n";
30
31       // Get the input again.
32       cout << "How many players do you wish per team? ";
33       cin >> teamPlayers;
34    }
35
36    // Get the number of players available.
37    cout << "How many players are available? ";
38    cin >> players;
39
40    // Validate the input.
41    while (players <= 0)
42    {
43       // Get the input again.
44       cout << "Please enter 0 or greater: ";
45       cin >> players;
46    }
```



Counters

Counters

- Counter: a variable that is incremented or decremented each time a loop repeats
- Can be used to control execution of the loop (also known as the *loop control variable*)
- Must be initialized before entering loop

A Counter Variable Controls the Loop in Program 5-6

Program 5-6

```
1  // This program displays a list of numbers and
2  // their squares.
3  #include <iostream>
4  using namespace std;
5
6  int main()
7  {
8     const int MIN_NUMBER = 1,    // Starting number to square
9               MAX_NUMBER = 10;   // Maximum number to square
10
11    int num = MIN_NUMBER;        // Counter
12
13    cout << "Number Number Squared\n";
14    cout << "-------------------------\n";
```

Continued…

A Counter Variable Controls the Loop in Program 5-6

```
15    while (num <= MAX_NUMBER)
16    {
17       cout << num << "\t\t" << (num * num) << endl;
18       num++; //Increment the counter.
19    }
20    return 0;
21 }
```

Program Output

```
Number Number Squared
---------------------
1           1
2           4
3           9
4           16
5           25
6           36
7           49
8           64
9           81
10          100
```

THANK YOU

QUESTIONS