

Computer Science I

Chapter 5 - Loops and Files

Hanoi University of Science and Technology

November 12, 2025



Slide contents are mainly based on the following documents:

- Tony Gaddis, Starting out with C++, Six Edition, Pearson, 2009
- Tony Gaddis, Power Points Slides:
<https://www.ums1.edu/~antognolij/gaddis.html>



The Increment and Decrement Operators

- `++` is the increment operator. It adds one to a variable.
`val++;` is the same as `val = val + 1;`
- `++` can be used before (prefix) or after (postfix) a variable:
`++val;` `val++;`



The Increment and Decrement Operators

- `--` is the decrement operator.
It subtracts one from a variable.
`val--;` is the same as `val = val - 1;`
- `--` can be used before (prefix) or after (postfix) a variable:
`--val;` `val--;`



The Increment and Decrement Operators

Program 5-1

```
1 // This program demonstrates the ++ and -- operators.
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     int num = 4;    // num starts out with 4.
8
9     // Display the value in num.
10    cout << "The variable num is " << num << endl;
11    cout << "I will now increment num.\n\n";
12
13    // Use postfix ++ to increment num.
14    num++;
15    cout << "Now the variable num is " << num << endl;
16    cout << "I will increment num again.\n\n";
17
18    // Use prefix ++ to increment num.
19    ++num;
20    cout << "Now the variable num is " << num << endl;
21    cout << "I will now decrement num.\n\n";
22
23    // Use postfix -- to decrement num.
24    num--;
25    cout << "Now the variable num is " << num << endl;
26    cout << "I will decrement num again.\n\n";
27
```



The Increment and Decrement Operators

Program 5-1 *(continued)*

```
28      // Use prefix -- to increment num.
29      --num;
30      cout << "Now the variable num is " << num << endl;
31      return 0;
32  }
```

Program Output

```
The variable num is 4
I will now increment num.

Now the variable num is 5
I will increment num again.

Now the variable num is 6
I will now decrement num.

Now the variable num is 5
I will decrement num again.

Now the variable num is 4
```



Prefix vs. Postfix

- `++` and `--` operators can be used in complex statements and expressions
- In prefix mode (`++val`, `--val`), the operator increments or decrements, then returns the value of the variable
- In postfix mode (`val++`, `val--`), the operator returns the value of the variable, then increments or decrements



Prefix vs. Postfix

```
int num, val = 12;  
cout << val++; // displays 12, val is now 13;  
cout << ++val; // sets val to 14, then displays it  
num = --val; // sets val to 13, stores 13 in num  
num = val--; // stores 13 in num, sets val to 12
```



Notes on Increment and Decrement

- Can be used in expressions:

```
result = num1++ + --num2;
```

- Must be applied to something that has a location in memory.

Cannot have:

```
result = (num1 + num2)++;
```

- Can be used in relational expressions:

```
if (++num > limit)
```

pre- and post-operations will cause different comparisons



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 `{ }`

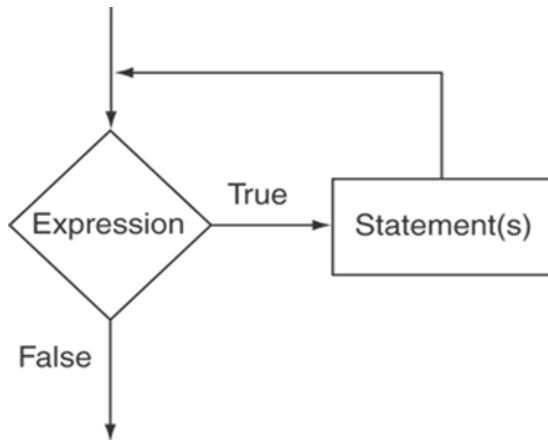


The while Loop – How It Works

- `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



while Loop example

Program 5-3

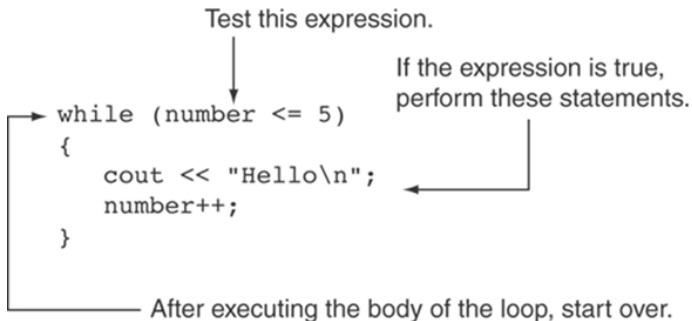
```
1  // This program demonstrates a simple while loop.
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7      int number = 1;
8
9      while (number <= 5)
10     {
11         cout << "Hello\n";
12         number++;
13     }
14     cout << "That's all!\n";
15     return 0;
16 }
```

Program Output

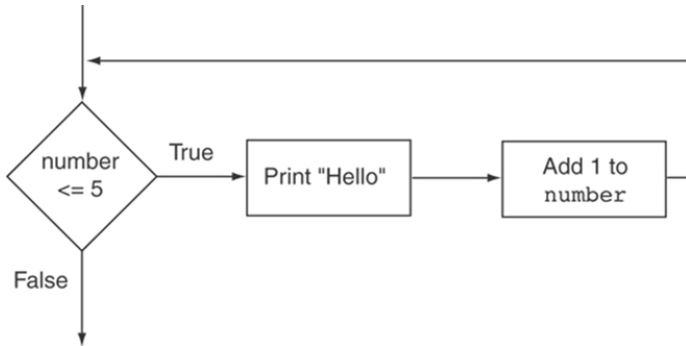
```
Hello
Hello
Hello
Hello
Hello
That's all!
```



while Loop example



while Loop example



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.
- 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

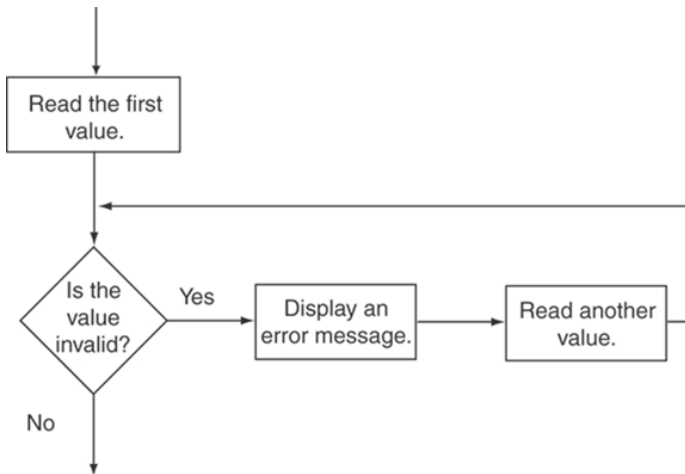


Input Validation Example

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



Input Validation Example



- 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



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";
```



Counters

```
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

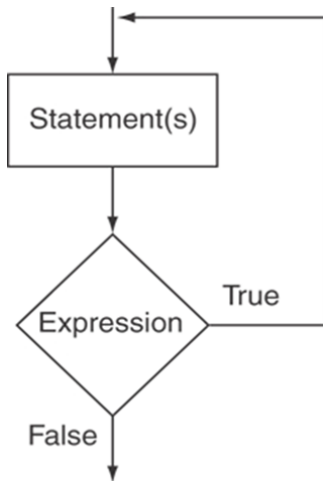


The do-while Loop

- do-while: a posttest loop – execute the loop, then test the expression
- General Format:
do
 statement; // or block in { }
while (expression);
- Note that a semicolon is required after (expression)



The do-while Loop



An Example do-while Loop

- ```
int x = 1;
do
{
 cout << x << endl;
} while(x < 0);
```
- Although the test expression is false, this loop will execute one time because do-while is a posttest loop.



# An Example do-while Loop

## Program 5-7

```
1 // This program averages 3 test scores. It repeats as
2 // many times as the user wishes.
3 #include <iostream>
4 using namespace std;
5
6 int main()
7 {
8 int score1, score2, score3; // Three scores
9 double average; // Average score
10 char again; // To hold Y or N input
11
12 do
13 {
14 // Get three scores.
15 cout << "Enter 3 scores and I will average them: ";
16 cin >> score1 >> score2 >> score3;
17
18 // Calculate and display the average.
19 average = (score1 + score2 + score3) / 3.0;
20 cout << "The average is " << average << ".\n";
21
22 // Does the user want to average another set?
23 cout << "Do you want to average another set? (Y/N) ";
24 cin >> again;
25 } while (again == 'Y' || again == 'y');
```



# An Example do-while Loop

## Program Output with Example Input Shown in Bold

Enter 3 scores and I will average them: **80 90 70** [Enter]

The average is 80.

Do you want to average another set? (Y/N) **y** [Enter]

Enter 3 scores and I will average them: **60 75 88** [Enter]

The average is 74.3333.

Do you want to average another set? (Y/N) **n** [Enter]



# do-while Loop Notes

- Loop always executes at least once
- Execution continues as long as expression is true, stops repetition when expression becomes false

- Challenging problem: 1 - 7
- Write a program that asks users to enter a positive integer  $n$  and calculates the factorial of  $n$  ( $n!$ )
- Write a program that asks users to enter a positive integer  $n$  and check if  $n$  is a prime number or not
- Write a program that asks users to enter a positive integer  $n$  and finds the reverse of  $n$ . For example, if  $n = 1234$ , then the reverse of  $n$  is 4321
- Write a program that asks users to enter two positive integers and determines the greatest common divisor of these integers
- Write a program that asks users to enter a positive integer  $n$  and a positive floating-point number  $x$ . Calculates the sum
$$S(n) = x + x^2 + x^3 + \dots + x^n$$

