4.3 More while examples

Students:
Section 4.4.4 is part of 2 assignments: CSC108 CH04.1-4.6 C4A

Please browse to this assignment through BlackboardLearn so zybooks knows where to send your activity. Learn more

Includes: CA
Due: 03/11/2025, 11:59 PM EDT

4.4 For loops

Basics

A loop commonly must iterate a specific number of times, such as 10 times. Though achievable with a while loop, that situation is so common that a special kind of loop exists. A **for loop** is a loop type with three parts: the keyword for, the loop variable initialization, a loop expression, and a loop variable update. A for loop describes iterating a specific number of times more naturally than a while loop.

Construct 4.4.1: For loop.

```
for (initialExpression; conditionExpression; updateExpression:
   // loop body

// Statements after the loop
```

Feedback?

PARTICIPATION ACTIVITY 4.4.1: For loops.

Captions

1. This while loop pattern with i = 0 before, loop expression i < 5, and loop body ending with i = i + 1, iterates 5 times: when i = 0, 1, 2, 3, and 4.
2. The pattern is so common that a special construct, a for loop, exists to collect the three parts in one place at the loop's top, improving readability and reducing errors.
3. Note that semicolons separate the three parts. No semicolon is needed at the end.

Feedback?

The statement i = i + 1 is so common that the language supports the shorthand **++i**, with ++ known as the ***increment operator***. Likewise, **--** is the ***decrement operator***. --means i = i - 1). As such, a standard way to loop N times is shown below.

Figure 4.4.1: A standard way to loop N times, using a for loop.

```
for (i = 0; i < N; ++i) {
   ...
}
```

Feedback?

PARTICIPATION ACTIVITY 4.4.2: For loops.

1) What are the values of i for each iteration of:

```
for (i = 0; i < 6; ++i) {
   ...
}
```

- 1, 2, 3, 4, 5
- 0, 1, 2, 3, 4, 5
- 0, 1, 2, 3, 4, 5, 6

2) How many times will this loop iterate?

```
for (i = 0; i < 8; ++i) {
   ...
}
```

- 7 times
- 8 times
- 9 times

3) Goal: Loop 10 times.

```
for (i = 0; _____; ++i) {
   ...
}
```

- i < 9
- i < 10
- i < 11

4) Goal: Loop 99 times.

```
for (i = 0; _____; ++i) {
   ...
}
```

- i < 99
- i <= 99
- i >= 99

5) Goal: Loop 20 times.

```
for (_____; i < 20; ++i) {
   ...
}
```

- i < 0
- i < 1

6) Goal: Loop numYears times (numYears is an int variable).

```
for (i = 0; _____; ++i) {
   ...
}
```

- numYears
- i <= numYears
- i < numYears

Feedback?

PARTICIPATION ACTIVITY 4.4.3: For loops.

Write for loops using the following form:

```
for (i = 0; i < 10; ++i) {
   ...
}
```

Note: Using a variable other than i is not accepted by this activity.

1) Complete the for loop to iterate 5 times. (Don't forget the semicolon).

```
for (i = 0;                 ++i) {
   ...
}
```

Check Show answer

2) Complete the for loop to iterate 7 times.

```
for (                       ++i) {
   ...
}
```

Check Show answer

3) Complete the for loop to iterate 500 times. (Don't forget the parenthesis).

```
for                         {
   ...
}
```

Check Show answer

4) Complete the for loop to iterate numDogs times. numDogs is an int variable.

```
for (i = 0;                 ++i) {
   ...
}
```

Check Show answer

Feedback?

*Note: Actually two increment operators exist: ++i (**pre-increment**, and i++ (**post-increment**). ++i increments before evaluating to a value, while i++ increments after. Ex: if i is 5, outputting ++i outputs 6, while outputting i++ outputs 5 (and then i becomes 6). This material primarily uses ++i for simplicity and safety, although many programmers use i++, especially in for loops.*

Example: Savings with interest

The following program outputs the amount of a savings account each year for 10 years, given an input initial amount and interest rate. A for loop iterates 10 times, such that each iteration represents one year, outputting that year's savings amount.

Figure 4.4.2: For loop: Savings interest program.

```
#include <iostream>
using namespace std;

int main() {
   double initialSavings;  // User-entered initial savings
   double interestRate;    // User-entered interest rate
   double curSavings;      // Current savings amount (interest)
   int i;                  // Loop variable

   cout << "Enter initial savings: ";
   cin >> initialSavings;

   cout << "Enter interest rate: ";
   cin >> interestRate;

   cout << endl << "Annual savings for 10 years:" << endl;

   curSavings = initialSavings;
   for (i = 1; i <= 10; ++i) {
      cout << "Y" << curSavings << endl;
      curSavings = curSavings + curSavings * interestRate;
   }

   return 0;
}
```

```
Enter initial savings: 1000
Enter interest rate: 0.05

Annual savings for 10 years:
Y1000
Y1050
Y1102.5
Y1157.6
Y1215.5
Y1276.3
Y1340.1
Y1407.1
Y1477.4
Y1551.3
Y1628.9
```

Feedback?

PARTICIPATION ACTIVITY 4.4.4: Savings interest program.

Consider the example above:

1) How many times does the for loop iterate?

- 9
- 10
- 12

2) During each iteration, the loop body's statements output the current savings amount, and then _____.

- increment i
- update curSavings

3) Can the input values change the number of loop iterations?

- Yes
- No

Feedback?

Example: Computing the average of a list of input values

The example below computes the average of an input list of integer values. The first input indicates the number of values in the subsequent list. That number controls how many times the subsequent for loop iterates.

Figure 4.4.3: Computing an average, with first value indicating list size.

```
#include <iostream>
using namespace std;

// Inputs average of list of integers
// First value indicates list size
// Ex: 4 10 5 8 3 indicates 10 + 5 + 8 + 3, or 26

int main() {
   int numValues;
   int valueSum = 0;
   int numValues;
   int i;

   cin >> numValues; // Gets number of values in list

   valueSum = 0;

   for (i = 0; i < numValues; ++i) {
      cin >> numValues; // Gets next value in list
      valueSum = valueSum + numValues;
   }

   cout << "Average: " << valueSum / numValues << endl;

   return 0;
}
```

```
4 10 5 8 3
Average: 5

...

4 -75 -50 30 40 80
Average: 8
```

Feedback?

PARTICIPATION ACTIVITY 4.4.5: Computing the average.

Consider the example above, with input 4 10 5 8 3. Note: The first input indicates the number of values in the subsequent list.

1) Before the loop is entered, what is valueSum?

Check Show answer

2) What is valueSum after the first iteration?

Check Show answer

3) What is valueSum after the second iteration?

Check Show answer

4) valueSum is 30 after the fourth iteration. What is numValues?

Check Show answer

5) For the following input, how many times will the for loop iterate?
7 1 -3 5 -14 15 -20 -40

Check Show answer

Feedback?

Choosing among for and while loops

Generally, a programmer uses a for loop when the number of iterations is known (like loop 5 times, or loop numItems times), and a while loop otherwise.

Table 4.4.1: Choosing between while and for loops: General guidelines (not strict rules though).

for	Number of iterations is computable before the loop, like iterating N times.
while	Number of iterations is not (easily) computable before the loop, like iterating until the input is 'q'.

Feedback?

PARTICIPATION ACTIVITY 4.4.6: While loops and for loops.

Choose the most appropriate loop type:

1) Iterate as long as user entered char c is not 'q'.

- while
- for

2) Iterate until the values of x and y are equal, where x and y are changed in the loop body.

- while
- for

3) Iterate 100 times.

- while
- for

Feedback?

CHALLENGE ACTIVITY 4.4.1: Enter the for loop's output.

5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0

```
for (i = 0; i < 5; ++i) {
   cout << 0;
}
```

This activity failed to load. Please try refreshing the page. If that fails, you might also try clearing your browser's cache.

If an issue persists, send feedback to zybooks support

CHALLENGE ACTIVITY 4.4.2: Read and format integers.

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

```
for (i = 0; i < 16; ++i) {
   cout << i << " ";
}
```

Start

numVal is read from input representing the value to be output. Complete the for loop to output numVal 8 times.

Click here for example

```
#include <iostream>
using namespace std;

int main() {
   int numVal;
   int i;

   cin >> numVal;

   for (i = 1; i <= numVal; ++i) {
      cout << numVal << endl;
   }

   return 0;
}
```

1 2 3

Check Next level

Feedback?

How was this section? Provide section feedback

Activity summary for assignment: CSC108 CH04.1-4.6 C4A

Due: 03/11/2025, 11:59 PM EDT

5 / 36 points

Please browse to this assignment through BlackboardLearn so zybooks knows where to send your activity. Learn more

Completion details