Compound Statement*

Kenneth Leroy Busbee

This work is produced by OpenStax-CNX and licensed under the
Creative Commons Attribution License 3.0†

Abstract

An explanation for the need of compound statements within some programming languages.

1 The Need for a Compound Statement

For illustration we will use the syntax for the **if then else** control structure within the C++ programming language. However this problem generally exists for all control structures within any language that requires the use of compound statements. The syntax is:

```
if (expression)
  statement;
else
  statement;
```

Within the C++ programming language there can be **only one statement listed as the action part of a control structure**. Often, we will want to do more than one statement. This problem is overcome by creating a **compound statement**. The brace symbols – the opening { and the closing } - are used to create a compound statement. For example:

```
if(expression)
  {
  statement;
  statement;
  }
else
  {
  statement;
  statement;
  }
```

*Version 1.4: Mar 5, 2012 6:25 am -0600
†http://creativecommons.org/licenses/by/3.0/

Because programmers often forget that they can have **only one statement listed as the action part of a control structure**; the C++ programming industry encourages the use of indentation (to see the action parts clearly) and the use of compound statements (**braces**), even if there is only one action. Thus:

```
if(expression)
  {
  statement;
  }
else
  {
  statement;
  }
```

By writing code in this manner, if the programmer modifies the code by adding more statements to either the action true or the action false; they will not introduce either compiler or logic errors. Using indentation and braces should become standard practice for C++ programmers and programmers in any other language that require the use of compound statements with the control structures.

2 Other Uses of a Compound Statement

"A compound statement is a unit of code consisting of zero or more statements. It is also known as a **block**. The compound statement allows a group of statements to become one single entry. You used a compound statement in your first program when you formed the body of the function main. All C++ functions contain a compound statement known as the function body.

A compound statement consists of an opening brace, optional declarations, definitions, and statements, followed by a closing brace. Although all three are optional, one should be present."[1]

3 Definitions

Definition 1: compound statement
A unit of code consisting of zero or more statements.

Definition 2: block
Another name for a compound statement.

[1] Behrouz A. Forouzan and Richard F. Gilberg, Computer Science A Structured Approach using C++ Second Edition (United States of America: Thompson – Brooks/Cole, 2004) 100.