

COMPILER DIRECTIVES*

Kenneth Leroy Busbee

This work is produced by OpenStax-CNX and licensed under the Creative Commons Attribution License 2.0[†]

Abstract

A general explanation of pre-processor directives and how they are evaluated by the compiler. Discussed are include and define.

1 General Discussion

A **compiler directive** is an instruction to the compiler to complete a task before formally starting to compile the program, thus they are sometimes called pre-processor directives. Among other items, during the pre-processor step the compiler is looking for compiler directives and processes them as they are encountered. After completing the tasks as directed, the compiler proceeds to its second step where it checks for syntax errors (violations of the rules of the language) and converts the source code into an object code that contains machine language instructions, a data area, and a list of items to be resolved when the object file is linked to other object files.

Within C++ the pound symbol or # as the first character of a line indicates that the next word is a directive (or command word) to be evaluated. The two most common compiler directives are:

1. **include** – with the item following include being the name of a file that is to be inserted at that place in the file. The files are often called "Header Files" because the include directive is normally inserted toward the top of the file (at the head) as one of the first items.
2. **define** – with the item followed by an identifier name and a value. This identifier name and value is stored by the compiler and when it encounters the identifier name in the program it substitutes the value for the identifier name.

In the following example the include directive is inserting a file that contains code from the **Input-Output Stream** library. This file contains necessary code to use cout and cin for sending data to the monitor or getting data from the keyboard.

```
#include <iostream>
```

In the next example the define directive is being used to handle a constant (called a defined constant).

Example 1: Substituting PI

```
#define PI 3.14159  
...Later on in the program when it encounters PI
```

*Version 1.7: Jan 15, 2010 6:58 am -0600

[†]<http://creativecommons.org/licenses/by/2.0/>

```
....it will replace or substitute PI with the value 3.14159
....For example:
area_circle = radius * radius * PI;
    would become:
area_circle = radius * radius * 3.14159;
```

Of note, compiler directives in C++ do not have a semi-colon after them. Within C++ programming instructions or statements end with a semi-colon, but not compiler directives.

2 Definitions

Definition 1: compiler directive

An instruction to the compiler to complete a task before formally starting to compile the program.

Definition 2: include

A compiler directive to insert the contents of a file into the program.