

Princeton University

COS 217: Introduction to Programming Systems

C Statements

Statement Type	Statement Syntax	Examples
Expression Statement	<i>expression</i> ;	<pre>i = 5; printf("Hello"); 5; /* valid, but nonsensical */</pre>
Declaration Statement	<i>modifiers datatype variable [= initialvalue][,variable [= initialvalue]]...;</i>	<pre>int i; int i, j; int i = 5, j = 6; const int i; static int i; extern int i;</pre>
Compound Statement (alias Block)	{ <i>statement statement ...</i> }	<pre>{ int i; i = 5; ... }</pre>
If Statement	<pre>if (<i>integerexpr</i>) <i>statement</i>; if (<i>pointerexpr</i>) <i>statement</i>;</pre>	<pre>if (i == 5) { <i>statement</i>; <i>statement</i>; }</pre>
Switch Statement	<pre>switch (<i>integerexpr</i>) { case <i>integerconstant</i>: <i>statements</i> case <i>integerconstant</i>: <i>statements</i> default: <i>statements</i> }</pre>	<pre>switch (i) { case 1: <i>statement</i>; break; case 2: <i>statement</i>; break; default: <i>statement</i>; }</pre>
While Statement	<pre>while (<i>integerexpr</i>) <i>statement</i> while (<i>pointerexpr</i>) <i>statement</i></pre>	<pre>while (i < 5) { <i>statement</i>; <i>statement</i>; }</pre>
DoWhile Statement	<pre>do <i>statement</i> while (<i>integerexpr</i>); do <i>statement</i> while (<i>pointerexpr</i>);</pre>	<pre>do { <i>statement</i>; <i>statement</i>; } while (i < 5);</pre>
For Statement	<pre>for (<i>initexpr</i>; <i>integerexpr</i>; <i>increxpr</i>) <i>statement</i> for (<i>initexpr</i>; <i>pointerexpr</i>; <i>increxpr</i>) <i>statement</i></pre>	<pre>for (i = 0; i < 5; i++) { <i>statement</i>; <i>statement</i>; }</pre>
Return Statement	<pre>return; return <i>expr</i>;</pre>	<pre>return; return i + 5;</pre>
Break Statement	<pre>break;</pre>	<pre>while (i < 5) { <i>statement</i>; if (j == 6) break; <i>statement</i>; }</pre>
Continue Statement	<pre>continue;</pre>	<pre>while (i < 5) { <i>statement</i>; if (j == 6) continue; <i>statement</i>; }</pre>
Goto Statement	<pre>goto <i>label</i>;</pre>	<pre>mylabel: ... goto mylabel; ...</pre>

Differences between C and Java:

Expression Statement:

- Java: Only expressions that have a side effect can be made into expression statements
- C: Any expression can be made into an expression statement
- Java: Has `final` variables
- C: Has `const` variables

Declaration Statement:

- Java: Compile-time error to use a local variable before specifying its value
- C: Run-time error to use a local variable before specifying its value

Compound Statement:

- Java: Declaration statements can be placed anywhere within compound statement
- C: Declaration statements must appear before any other type of statement within compound statement

If Statement

- Java: Controlling `expr` must be of type boolean
- C: Controlling `expr` must be of some integer type or a pointer (0 => FALSE, non-0 => TRUE)

While Statement

- Java: Controlling `expr` must be of type boolean
- C: Controlling `expr` must be of some integer type or a pointer (0 => FALSE, non-0 => TRUE)

DoWhile Statement

- Java: Controlling `expr` must be of type boolean
- C: Controlling `expr` must be of some integer type or a pointer (0 => FALSE, non-0 => TRUE)

For Statement

- Java: Controlling `expr` must be of type boolean
- C: Controlling `expr` must be of some integer type or a pointer (0 => FALSE, non-0 => TRUE)
- Java: Can declare loop control variable in `initexpr`
- C: Cannot declare loop control variable in `initexpr`

Break Statement

- Java: Also has "labeled `break`" statement
- C: Does not have "labeled `break`" statement

Continue Statement

- Java: Also has "labeled `continue`" statement
- C: Does not have "labeled `continue`" statement

Goto Statement

- Java: Not provided
- C: Provided (but don't use it!)