

## C++ Quick Reference (continued)

### The for Loop

#### Form:

```
for (initialization; test; update)
    statement;

for (initialization; test; update)
{
    statement;
    statement;
}
```

#### Example:

```
for (count = 0; count < 10; count++)
    cout << count << endl;

for (count = 0; count < 10; count++)
{
    cout << "The value of count is ";
    cout << count << endl;
}
```

### The switch/case Construct

#### Form:

```
switch (integer-expression)
{
    case integer-constant:
        statement(s);
        break;
    case integer-constant:
        statement(s);
        break;
    default :
        statement;
}
```

#### Example:

```
switch (choice)
{
    case 0 :
        cout << "You selected 0.\n";
        break;
    case 1 :
        cout << "You selected 1.\n";
        break;
    default :
        cout << "You did not select 0 or 1.\n";
}
```

### Using cout

Requires <iostream> header file.

#### Commonly used stream manipulators

Name	Description
endl	advances output to the beginning of the next line.
fixed	sets fixed point notation
left	sets left justification
right	sets right justification
setprecision	sets the number of significant digits
setw	sets field width
showpoint	forces decimal point & trailing zeros to display

#### Example:

```
cout << setprecision(2) << fixed
      << left << x << endl;
```

#### Member functions for output formatting

Name	Description
.precision	sets the number of significant digits
.setf	sets one or more ios flags
.unsetf	clears one or more ios flags
.width	sets field width

#### Example:

```
cout.precision(2);
```

### Using cin

Requires <iostream> header file

#### Commonly used stream manipulators

Name	Description
setw	sets field width

#### Member functions for input formatting

Name	Description
.getline	reads a line of input
.get	reads a character
.ignore	ignores the last character entered
.width	sets field width

### Some Commonly Used Library Functions

#### Name Description

(The following require <cstdlib>)

atof	Converts C-string to float
atoi	Converts C-string to int
atol	Converts C-string to long int
rand	Generates a pseudo-random number
srand	Sets seed value for random numbers

(The following require <cctype>)

islower	Returns true if char argument is lowercase
isupper	Returns true if char argument is uppercase
tolower	Returns the lowercase equivalent of the char argument
toupper	Returns the uppercase equivalent of the char argument

(The following require <cmath>)

pow	Raises a number to a power
sqrt	Returns square root of a number

(The following require <cstring>)

strcat	Appends a C-string to another C-string
strcpy	Copies a C-string
strlen	Returns the length of a C-string

## C++ Quick Reference

<i>C++ Data Types</i>	
Data Type	Description
char	Character
unsigned char	Unsigned Character
int	Integer
short int	Short integer
short	Same as short int
unsigned short int	Unsigned short integer
unsigned short	Same as unsigned short int
unsigned int	Unsigned integer
unsigned	Same as unsigned int
long int	Long integer
long	Same as long int
unsigned long int	Unsigned long integer
unsigned long	Same as unsigned long int
float	Single precision floating point
double	double precision floating point
long double	Long double precision floating point

<i>Forms of the if Statement</i>	
<b>Simple if</b>	<b>Example</b>
if ( <i>expression</i> ) <i>statement</i> ;	if (x < y) x++;
<b>if/else</b>	<b>Example</b>
if ( <i>expression</i> ) <i>statement</i> ; else <i>statement</i> ;	if (x < y) x++; else x--;
<b>if/else if</b>	<b>Example</b>
if ( <i>expression</i> ) <i>statement</i> ; else if ( <i>expression</i> ) <i>statement</i> ; else <i>statement</i> ;	if (x < y) x++; else if (x < z) x--; else y++;
<b>To conditionally-execute more than one statement, enclose the statements in braces:</b>	
<b>Form</b>	<b>Example</b>
if ( <i>expression</i> ) { <i>statement</i> ; <i>statement</i> ; }	if (x < y) { x++; cout << x; }

**Web Sites**  
 For the *Starting Out with C++ Series*  
[aw.com/gaddisbooks](http://aw.com/gaddisbooks)  
 For Addison-Wesley Computing  
[aw.com/computing](http://aw.com/computing)

<i>Commonly Used Operators</i>	
<b>Assignment Operators</b>	
=	Assignment
+=	Combined addition/assignment
-=	Combined subtraction/assignment
*=	Combined multiplication/assignment
/=	Combined division/assignment
%=	Combined modulus/assignment
<b>Arithmetic Operators</b>	
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus (remainder)
<b>Relational Operators</b>	
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
!=	Not equal to
<b>Logical Operators</b>	
&&	AND
	OR
!	NOT
<b>Increment/Decrement</b>	
++	Increment
--	Decrement

**Conditional Operator ?:**  
**Form:**  
*expression* ? *expression* : *expression*  
**Example:**  
 x = a < b ? a : b;  
**The statement above works like:**  
 if (a < b)  
  x = a;  
 else  
  x = b;

**The while Loop**  
**Form:**  
 while (*expression*)  
  *statement*;

**Example:**  
 while (x < 100)  
  cout << x++ << endl;

while (*expression*)  
{  
  *statement*;  
  *statement*;  
}

while (x < 100)  
{  
  cout << x << endl;  
  x++;  
}

**The do-while Loop**  
**Form:**  
 do  
  *statement*;  
while (*expression*);

**Example:**  
 do  
  cout << x++ << endl;  
while (x < 100);

do  
{  
  *statement*;  
  *statement*;  
} while (*expression*);

do  
{  
  cout << x << endl;  
  x++;  
} while (x < 100);