Because the precedence of the conditional operator is just above assignment, parentheses are not necessary. However, parentheses often are used to make clear what is being tested for.

The type of the conditional expression

expr1 ? *expr2* : *expr3*

is determined by both *expr2* and *expr3*. If they are of different types, then the usual conversion rules are applied. Note carefully that the type of the conditional expression does not depend on which of the two expressions *expr2* or *expr3* is evaluated. The conditional operator ?: has precedence just above the assignment operators, and it associates from right to left.

Declarations and initializations			

char a = 'a', b = 'b'; /* a has decimal value 97 */
int i = 1, j = 2;
double x = 7.07;			

Expression	Equivalent expression	Value	Type
i == j ? a – 1 : b + 1	(i == j) ? (a – 1) : (b + 1)	99	int
j % 3 == 0 ? i + 4 : x	((j % 3) == 0) ? (i + 4) : x	7.07	double
j % 3 ? i + 4 : x	(j % 3) ? (i + 4) : x	5.0	double

Summary

1 Relational, equality, and logical expressions have the int value 0 or 1.

2 The negation operator ! is unary. A negation expression such as !a has the int value 0 or 1. Remember: !!a and a need not have the same value.

3 A chief use of relational, equality, and logical expressions is to test data to affect flow of control.

4 Automatic type conversions can occur when two expressions are compared that are the operands of a relational, equality, or logical operator.

5 The grouping construct { ····· } is a compound statement. It allows enclosed statements to be treated as a single unit.

6 An if statement provides a way of choosing whether or not to execute a statement.

7 The else part of an if-else statement associates with the nearest available if. This resolves the "dangling else" problem.

8 The while, for, and do statements provide for the iterative execution of code. The body of a do statement executes at least once.

9 To prevent an unwanted infinite loop, the programmer must make sure that the expression controlling the loop eventually becomes zero. The miscoding of controlling expressions is a common programming error.

10 The programmer often has to choose between the use of a while or a for statement. In situations where clarity dictates that both the control and the indexing be kept visible at the top of the loop, the for statement is the natural choice.

11 The comma operator is occasionally useful in for statements. Of all the operators in C, it has the lowest priority.

12 The four statement types

goto break continue return

cause an unconditional transfer of flow of control. Their use should be minimized.

13 goto's are considered harmful to good programming. Avoid them.

14 The switch statement provides a multiway conditional branch. It is useful when dealing with a large number of special cases.

Exercises

1 Give equivalent logical expressions of the following without negation:

```
!(a > b)
!(a <= b && c <= d)
!(a + 1 == b + 1)
!(a < 1 || b < 2 && c < 3)
```