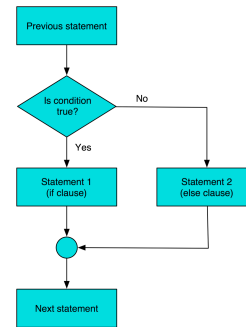


Conditional Statements

CSE 130: Introduction to C Programming
Spring 2005

1

Flow of Control



2

Conditions

- Conditional statements execute a test to determine which path to follow
- This test consists of an expression that is evaluated
- Normally, this expression compares two or more values

3

True and False Values

- Any expression with a non-zero value is considered to be true
 - Ex. 1, 3.14159, -23
- An expression is only false if its value is 0
- Common programming error: using '=' (assignment) instead of '==' (equality)
 - Ex. `if (x = 5)`

4

Relational Operators

Operator	Meaning	Example
<	Less than	<code>age < 30</code>
>	Greater than	<code>height > 6.2</code>
<=	Less than or equal to	<code>taxable <= 20000</code>
>=	Greater than/equal to	<code>temp >= 98.6</code>
==	Equal to	<code>grade == 100</code>
!=	Not equal to	<code>number != 250</code>

5

Logical Operators

- && (logical AND)
 - true only if both operands are true
- || (logical OR)
 - true if at least one operand is true
- ! (logical NOT)
 - true if operand is false and vice versa

6

Expression Examples

- Assume $a = 5, b = 2, c = 4, d = 6$
- What are the values of these expressions?
 - $a > b$
 - $d \% b == c \% b$
 - $!(a * b)$
 - $!(c \% b * a)$
 - $d \% b * c > 5 || c \% b * d < 7$

7

The `if` Statement

- General form:

```
if ( condition )  
    statement (or block of statements) to be  
    executed if condition is true
```
- Ex.

```
if (length < 2)  
    printf("Too short!\n");
```

8

Compound Statements

- `if` and `else` only execute a single following statement
- We can get around this by enclosing multiple statements in braces
- The resulting block is called a compound statement
- Style suggestion: always use braces around the body of an `if` or `else` clause

9

```
if (key == 'F') {  
    contemp = (5.0/9.0) * (intemp - 32.0);  
    printf("Converted to Celsius\n");  
}  
else {  
    contemp = (9.0/5.0) * intemp + 32.0;  
    printf("Converted to Fahrenheit.\n");  
}
```

10

Empty Statements

- A semicolon by itself is a valid (but non-functional) statement
- Common mistake: putting a semicolon immediately after an `if` statement:

```
if (x > 5);  
    printf("x greater than 5!\n");
```

11

The `if-else` Statement

- Select one of two possible execution paths, based on the result of a comparison
- General format:

```
if ( expression )  
    statement block 1  
else  
    statement block 2
```

12

Nested if Statements

- A statement block may contain another if statement

- Ex.

```
if (income > 25000)
    if (deductions < 3500)
        tax_rate = 1.035;
```

13

if-else Chains

```
if (expression_1)
    statement_1
else if (expression_2)
    statement_2
else
    statement_3
```

14

The switch Statement

- Long if-else chains can be unwieldy to read and maintain
- The switch statement generalizes the if-else statement
- Specifies different actions (cases) depending on the value of an integer expression

15

The switch Statement

```
switch ( expression )
{
    case value : statement(s)
    case value : statement(s)
    ...
}
```

16

switch Execution

1. The integer expression is evaluated
 2. Execution jumps to the case statement whose label matches the expression value
 3. Execution continues until a break statement is encountered (or switch ends)
- Common error: failure to include break

17

switch Example

```
switch (labSection)
{
    case 1: printf("Section 01\n");
            break;
    case 2: printf("Section 02\n");
            break;
    default: printf("Invalid section\n");
}
}
```

18

The default Case

- The `default` label denotes statements to be executed if no case matches the expression value
- `default` is usually the last label in a switch statement
- Ex. **default:** `statement(s)`

19

Boolean Variables

- Modern versions of C include a `bool` type
- This type is defined in `<stdbool.h>`
- A `bool` variable can have two values: `true` and `false`
 - Ex. `bool isPrime = true;`

20

The Conditional Operator

- General form:
`expr_1 ? expr_2 : expr_3`
- If `expr_1` is true, the conditional statement's value is that of `expr_2`; otherwise, its value is that of `expr_3`
- This operator can be confusing to look at

21

Equivalent Code

```
if (y < z)
    x = y;
else
    x = z;

x = (y < z) ? y : z;
```

22

Next Time

- Arrays
- Functions

23