FLOW CONTROL : syntax, examples

CONTROL STATEMENT SYNTAX

There are several control statements available in C. They are; 'while', 'for', 'do', 'if', conditional, 'switch', and the infamous 'goto'. Syntax for these statements are:

```
while( expr )
  statement

for( init; test; update )
  statement

do                                    // statement executed at least once
  statement
while( expr );

if( expr )
  statement
else if( expr )
  statement
else
  statement

expr ? expr : expr                  // conditional

switch( expr )

  case const1: statement
               break;               // without break, control
  case const2: statement             // continues to next
               break;               // statement.
  case const3: statement
               break;
  default    : statement


goto label;
.
.
.
label:statement
```

FLOW CONTROL EXAMPLES

The following program provides some basic examples of each control statement. One of the more common typos is to use commas instead of semi-colons in 'for' statements. Although the goto statement is provided in C, it is usually avoided and most books advise against its use. The conditional statement is shorthand for an if-then statement, but it can decrease code readability (mainly because it is rarely used).

```
// control.cp
#include <iostream.h>

void main()

  int a = 1;
  int b = 2;
  int i = 1;
  cout << "start while loop" << endl;     // while
  while( i<4 )

    cout << "i = " << i << endl;
    i++;


  cout << "start for loop" << endl;        // for
  for( i = 1; i<4; i++ )
    cout << "i = " << i << endl;

  cout << "start do loop" << endl;          // do
  i = 1;
  do

    cout << "i = " << i << endl;
    i++;

  while( i<4 );

  cout << "start if" << endl;                  // if
  i = 2;
  if( i==a )
    cout << "answer is a" << endl;
  else if( i==b )
    cout << "answer is b" << endl;
  else
    cout << "neither a or b" << endl;

  cout << "conditional" << endl;             // conditional
  i = 10;
  i = ( i>b ) ? b : i;
  cout << "i = " << i << endl;

  cout << "switch statement" << endl;     // switch
  switch( i )

    case 1:   cout << "one\n";
              break;
    case 2:   cout << "two\n";
              break;
    case 3:   cout << "three\n";
              break;
    default: cout << "error\n";


  cout << "goto statement" << endl;        // goto
  if( i==b )
    goto skip; // functional scope
  cout << "zero\n";
  cout << "one\n";
  skip:
  cout << "two\n";
  cout << "three\n";

// end control.cp
```