Lecture-2 Loops and Conditionals

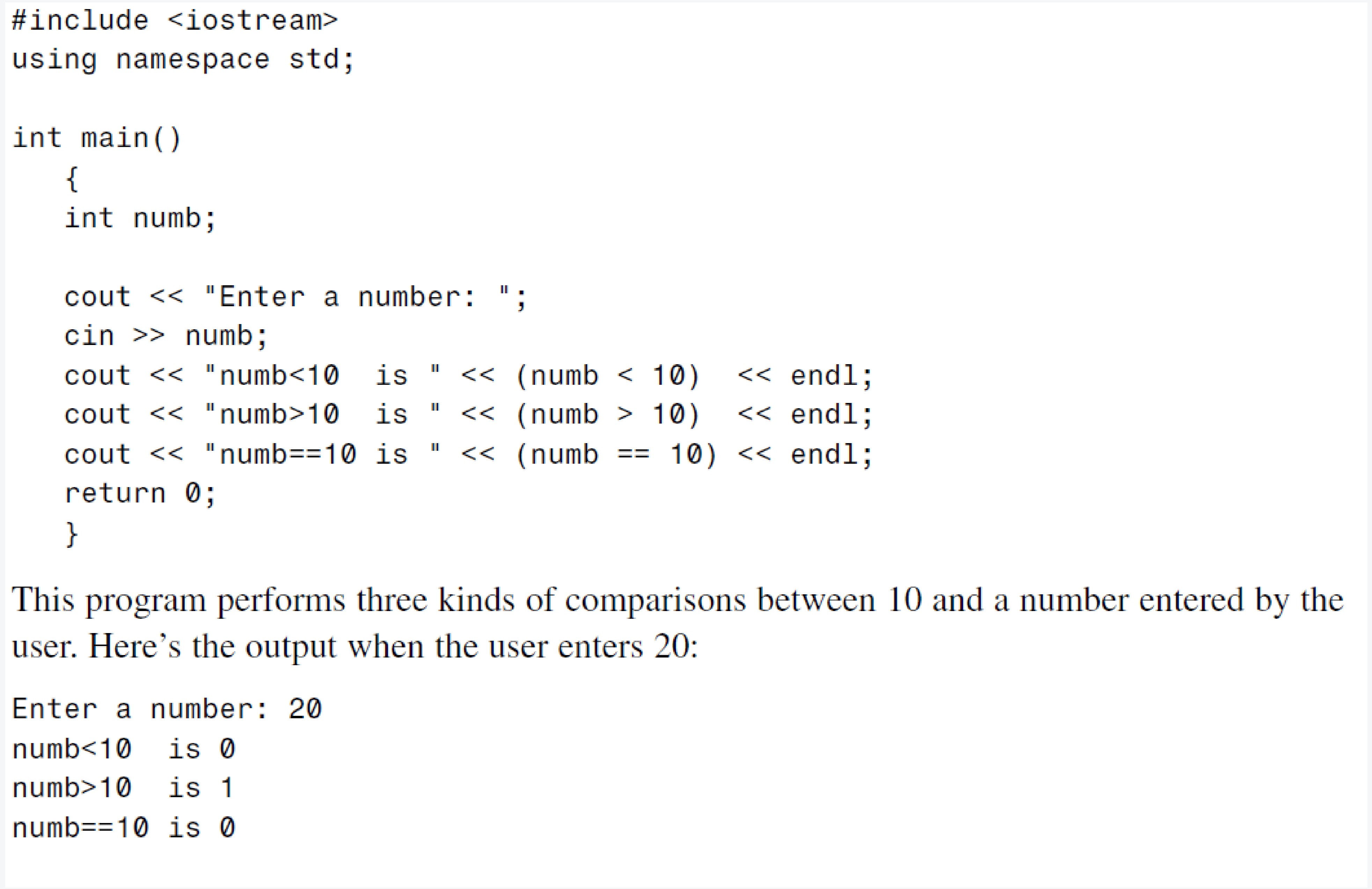
Agenda

01.

Relational Operators

Relational Operators

- A relational operator compares two values.
- The comparison involves such relationships as equal to, less than, and greater than.
- The result of the comparison is true or false.

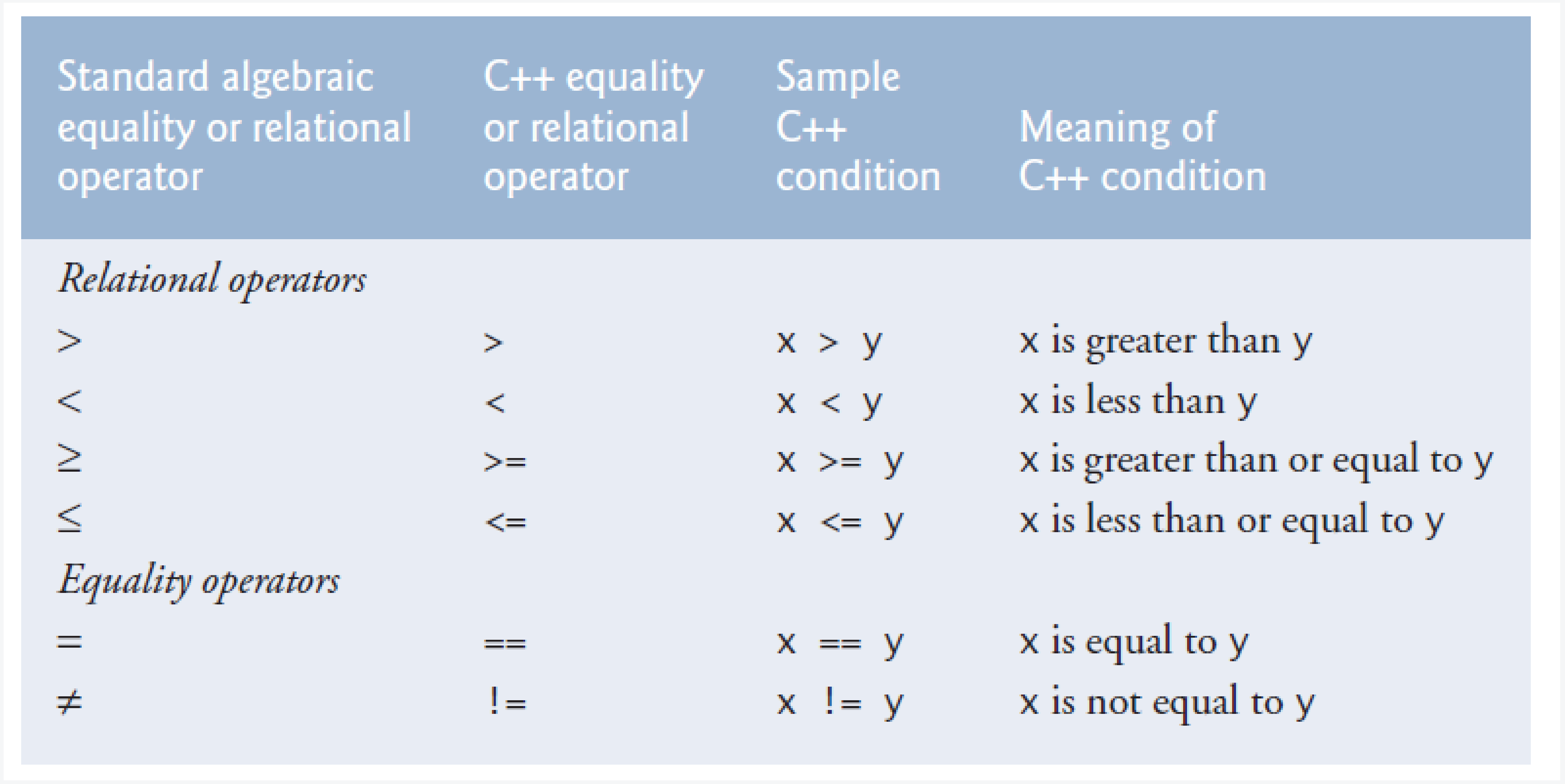


```
#include <iostream>
using namespace std;

int main()
   {
   int numb;

   cout << "Enter a number: ";
   cin >> numb;
   cout << "numb<10  is " << (numb < 10)  << endl;
   cout << "numb>10  is " << (numb > 10)  << endl;
   cout << "numb==10 is " << (numb == 10) << endl;
   return 0;
   }
```

This program performs three kinds of comparisons between 10 and a number entered by the user. Here's the output when the user enters 20:

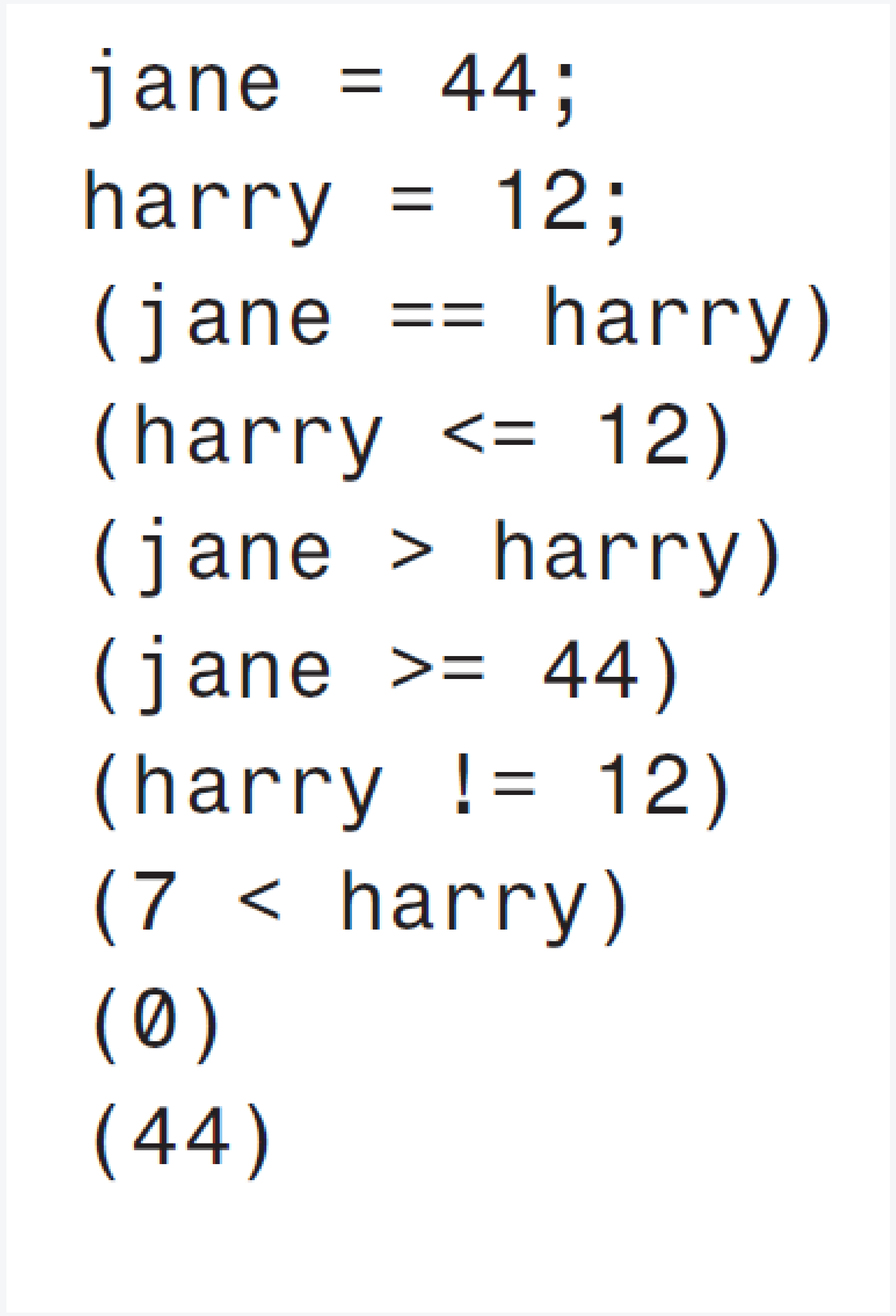
```
Enter a number: 20
numb<10  is 0
numb>10  is 1
numb==10 is 0
```

Relational Operators



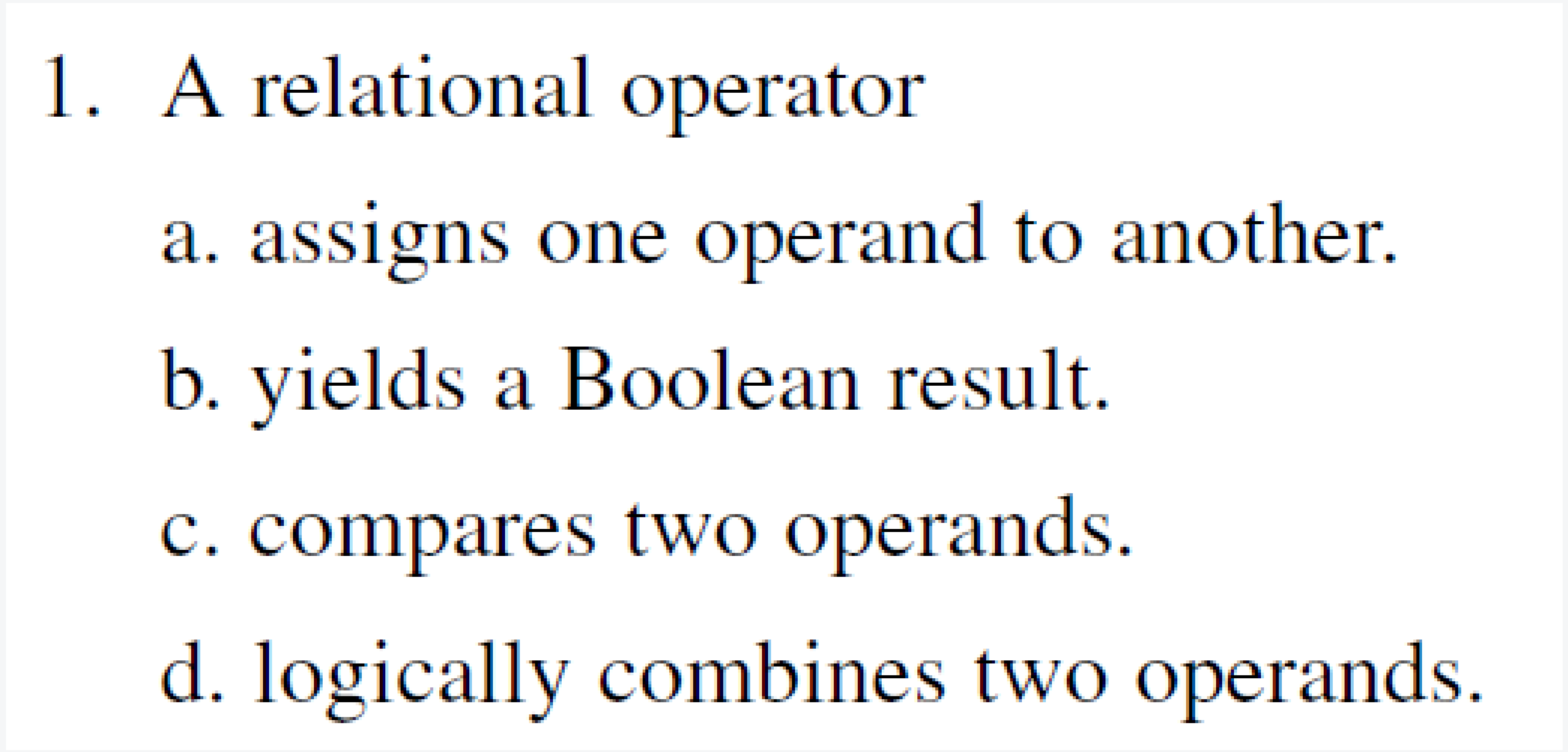
Standard algebraic equality or relational operator	C++ equality or relational operator	Sample C++ condition	Meaning of C++ condition
Relational operators			
$>$	`>`	`x > y`	x is greater than y
$<$	`<`	`x < y`	x is less than y
$\geq$	`>=`	`x >= y`	x is greater than or equal to y
$\leq$	`<=`	`x <= y`	x is less than or equal to y
Equality operators			
$=$	`==`	`x == y`	x is equal to y
$\neq$	`!=`	`x != y`	x is not equal to y

Relational Operators



```
jane = 44;
harry = 12;
(jane == harry)
(harry <= 12)
(jane > harry)
(jane >= 44)
(harry != 12)
(7 < harry)
(0)
(44)
```

Relational Operators

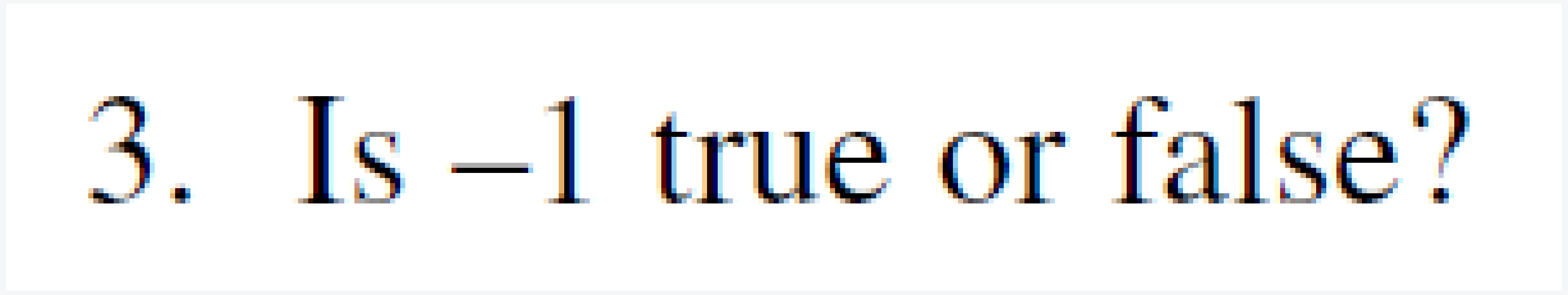


1. A relational operator
   a. assigns one operand to another.
   b. yields a Boolean result.
   c. compares two operands.
   d. logically combines two operands.

Relational Operators

2. Write an expression that uses a relational operator to return true if the variable `george` is not equal to `sally`.

Relational Operators



3. Is –1 true or false?

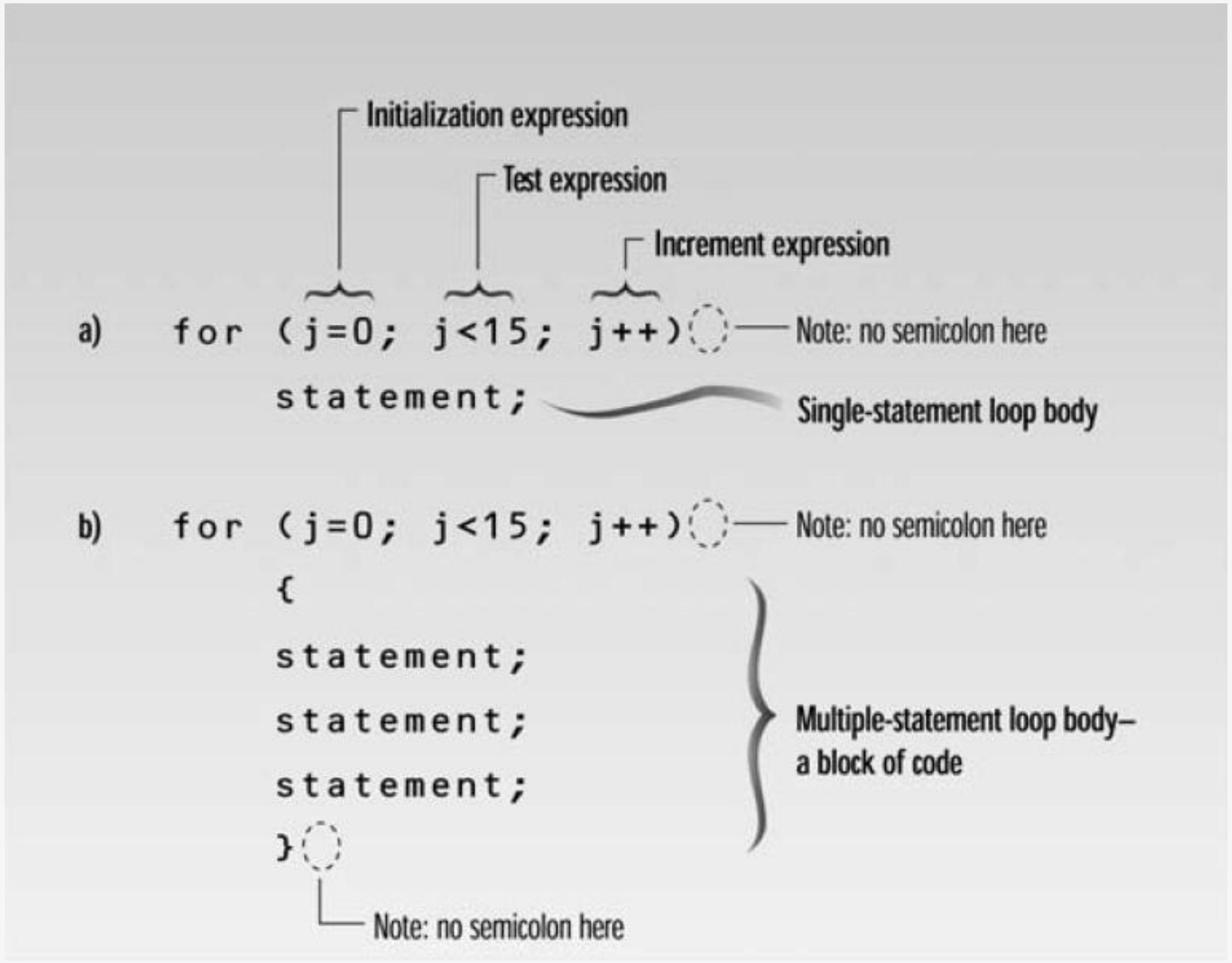
02.

Loops

Loops

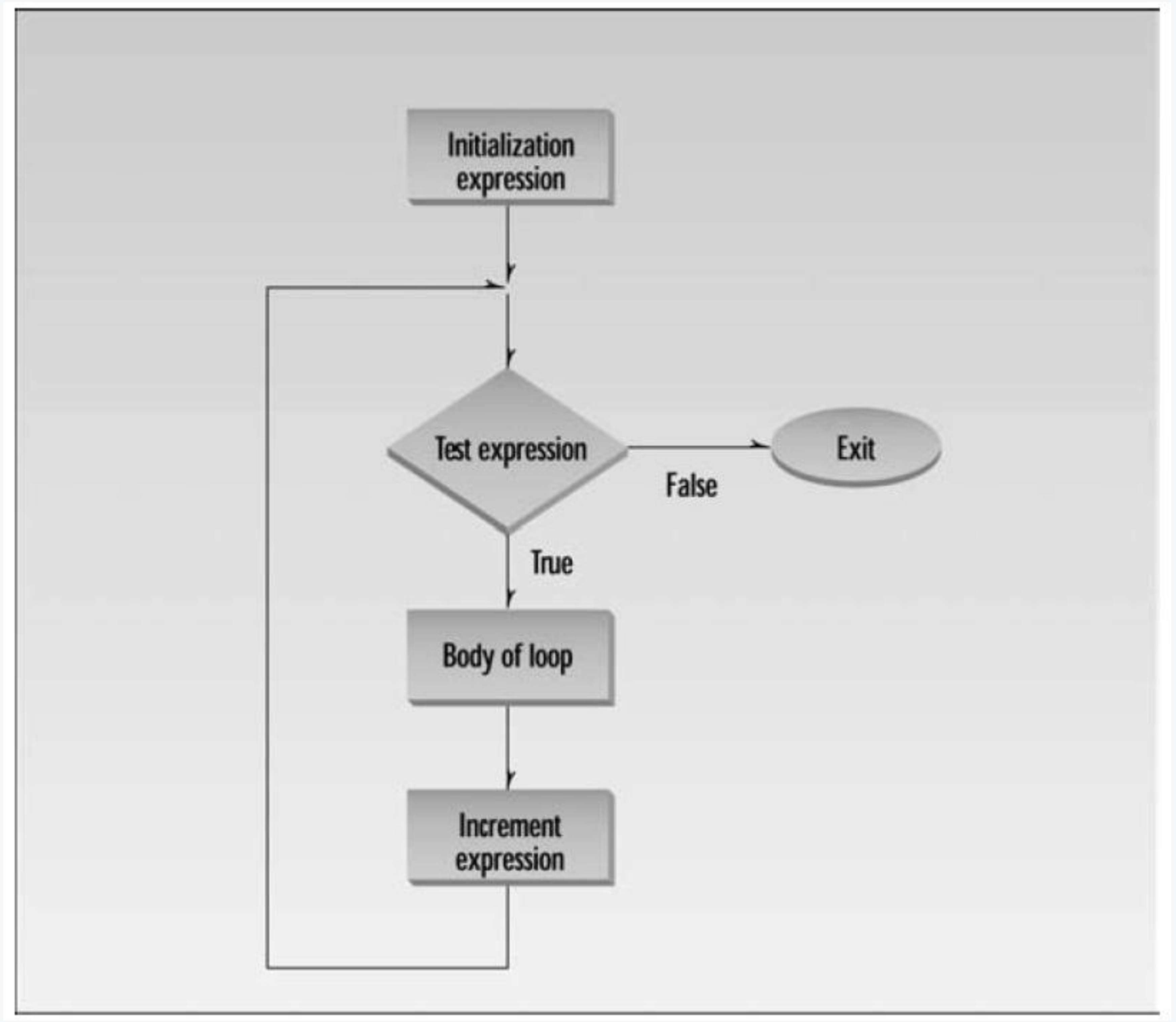
- Loops cause a section of your program to be repeated a certain number of times.
- The repetition continues while a condition is true. When the condition becomes false, the loop ends and control passes to the statements following the loop.

For Loop



```
a)    for (j=0; j<15; j++)
          statement;

b)    for (j=0; j<15; j++)
          {
          statement;
          statement;
          statement;
          }
```

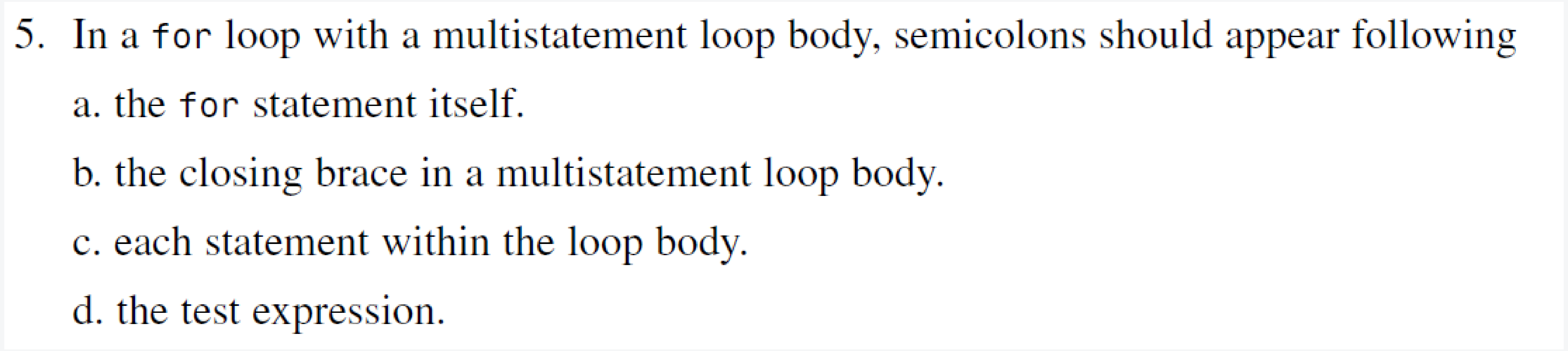
For Loop


Initialization expression
Test expression
False
Exit
True
Body of loop
Increment expression

For Loop

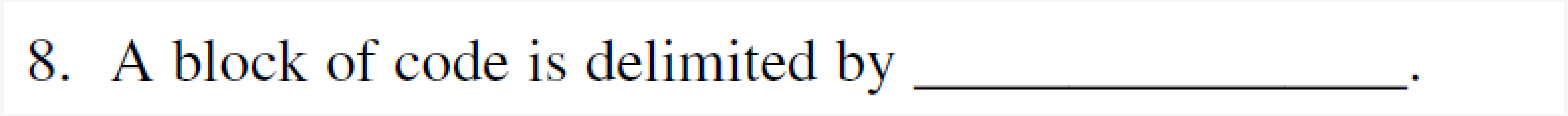
4. Name and describe the usual purpose of three expressions in a `for` statement.

For Loop



5. In a `for` loop with a multistatement loop body, semicolons should appear following
   a. the `for` statement itself.
   b. the closing brace in a multistatement loop body.
   c. each statement within the loop body.
   d. the test expression.

For Loop



8. A block of code is delimited by ________________.

For Loop

```
#include <iomanip>                          //for setw
using namespace std;

int main()
   {
   int numb;                                //define loop variable

   for(numb=1; numb<=10; numb++)            //loop from 1 to 10
      {
      cout << setw(4) << numb;              //display 1st column
      int cube = numb*numb*numb;            //calculate cube
      cout << setw(6) << cube << endl;      //display 2nd column
      }
   return 0;
   }
```

```
   1     1
   2     8
   3    27
   4    64
   5   125
   6   216
   7   343
   8   512
   9   729
  10  1000
```

For Loop

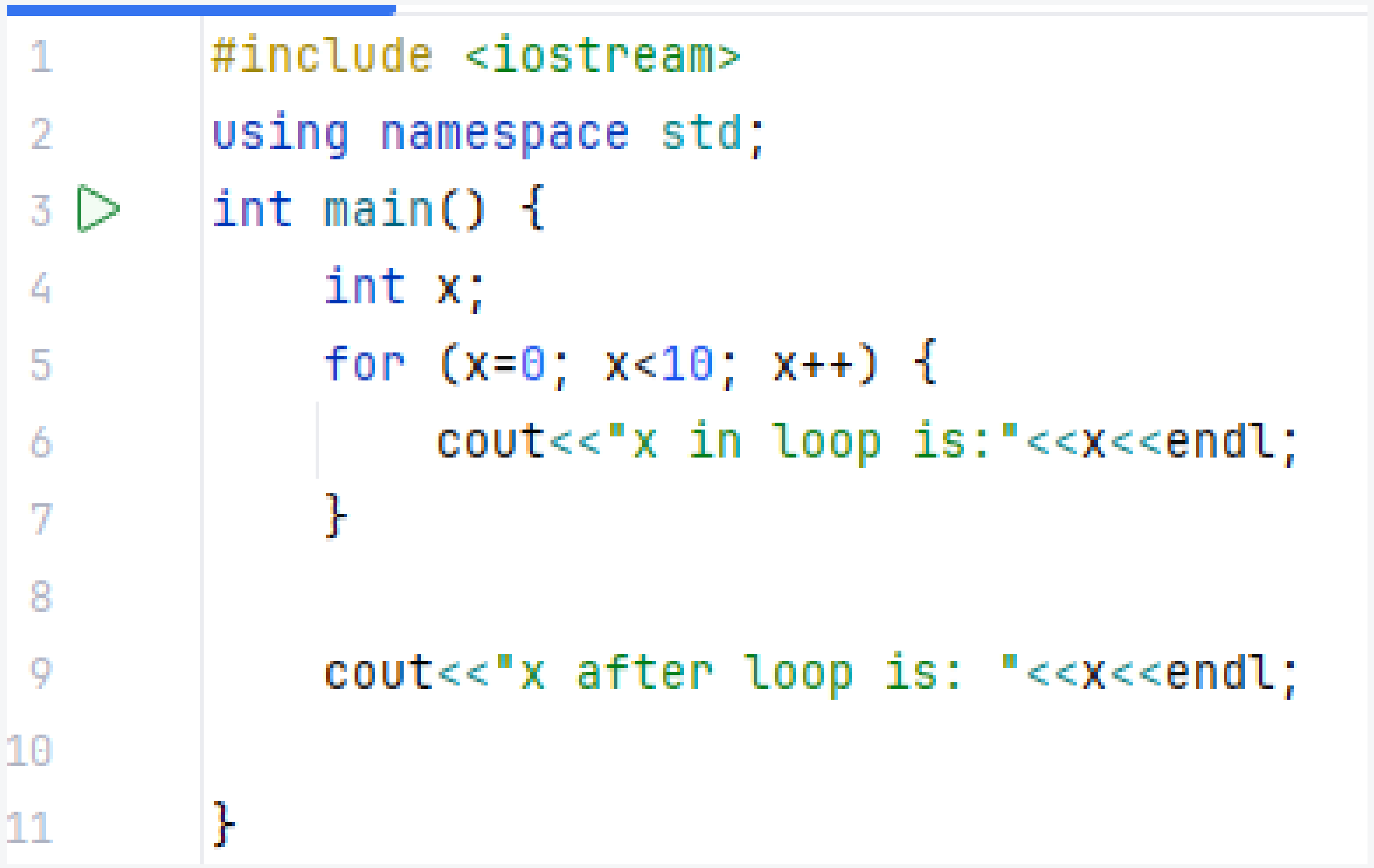
```
#include <iostream>
using namespace std;

int main()
   {
   unsigned int numb;
   unsigned long fact=1;              //long for larger numbers

   cout << "Enter a number: ";
   cin >> numb;                       //get number


 for(int j=numb; j>0; j--)           //multiply 1 by
    fact *= j;                        //numb, numb-1, ..., 2, 1
 cout << "Factorial is " << fact << endl;
 return 0;
 }
```

```
Enter a number: 10
Factorial is 3628800
```

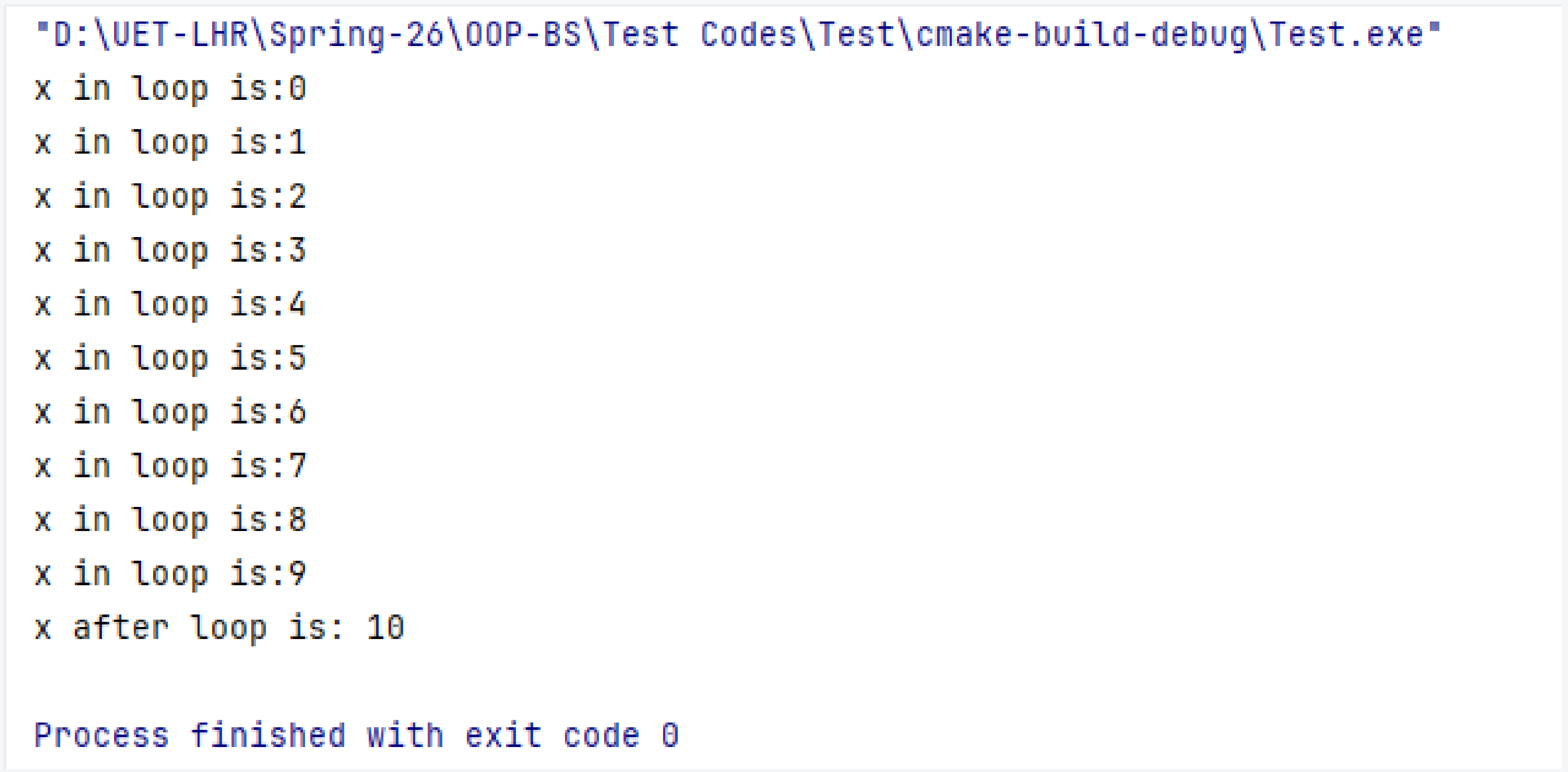
For Loop

```
for( j=0, alpha=100; j<50; j++, beta-- )
   {
   // body of loop
   }
```

For Loop

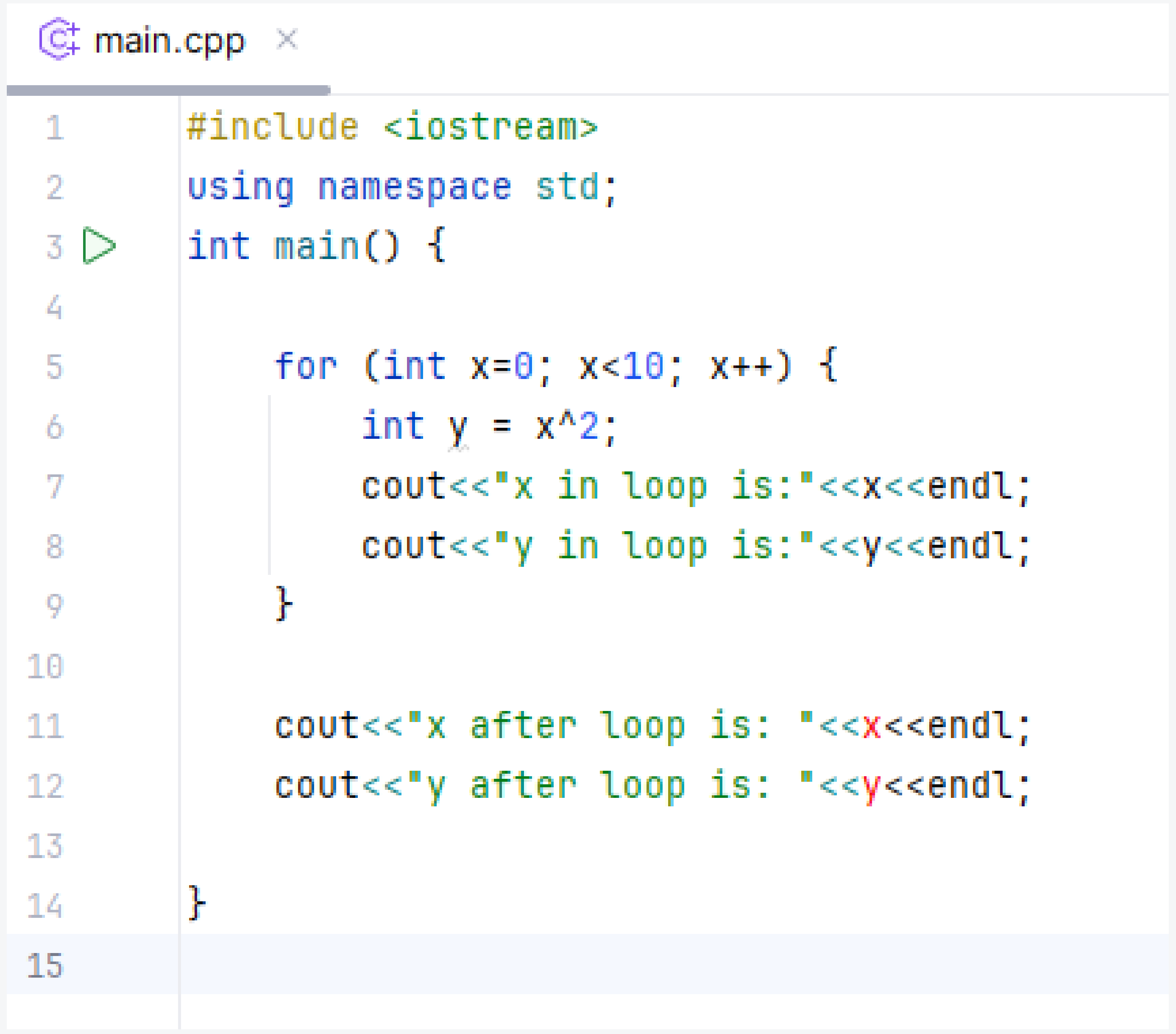


```cpp
#include <iostream>
using namespace std;
int main() {
    int x;
    for (x=0; x<10; x++) {
        cout<<"x in loop is:"<<x<<endl;
    }

    cout<<"x after loop is: "<<x<<endl;

}
```



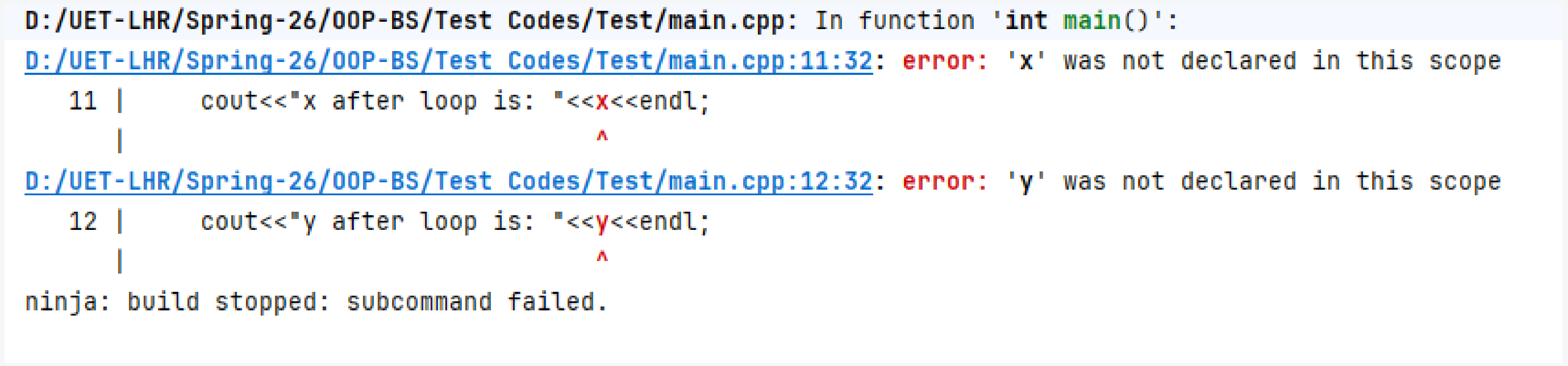
```
"D:\UET-LHR\Spring-26\OOP-BS\Test Codes\Test\cmake-build-debug\Test.exe"
x in loop is:0
x in loop is:1
x in loop is:2
x in loop is:3
x in loop is:4
x in loop is:5
x in loop is:6
x in loop is:7
x in loop is:8
x in loop is:9
x after loop is: 10

Process finished with exit code 0
```

For Loop



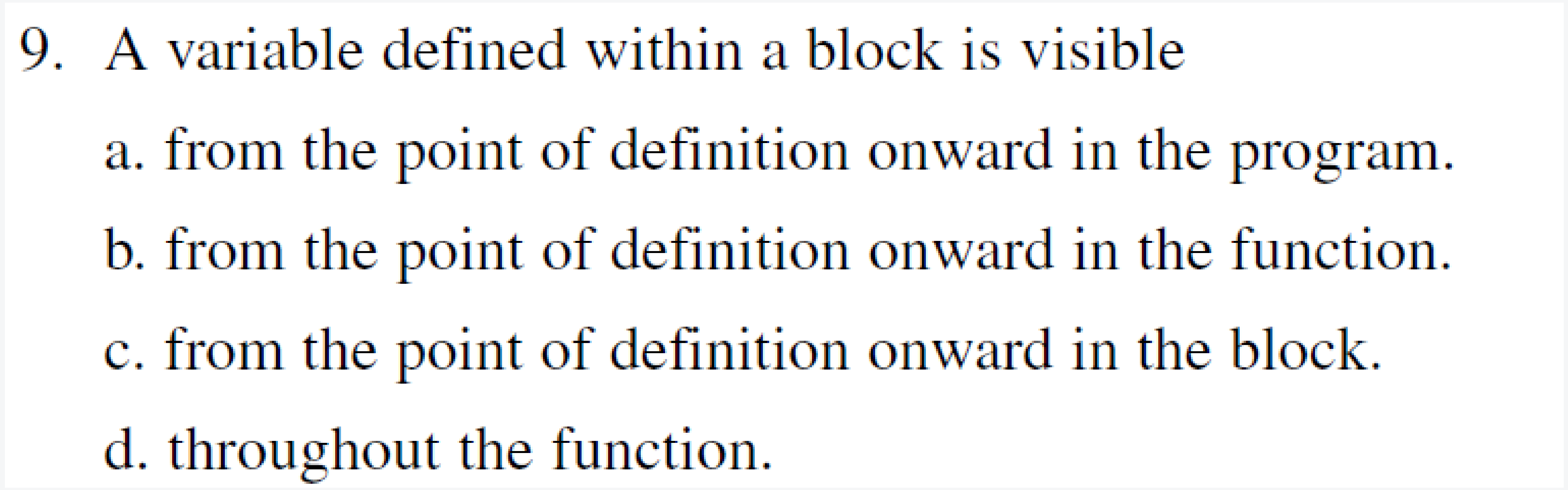
main.cpp

```cpp
#include <iostream>
using namespace std;
int main() {

    for (int x=0; x<10; x++) {
        int y = x^2;
        cout<<"x in loop is:"<<x<<endl;
        cout<<"y in loop is:"<<y<<endl;
    }

    cout<<"x after loop is: "<<x<<endl;
    cout<<"y after loop is: "<<y<<endl;

}
```



```
D:/UET-LHR/Spring-26/OOP-BS/Test Codes/Test/main.cpp: In function 'int main()':
D:/UET-LHR/Spring-26/OOP-BS/Test Codes/Test/main.cpp:11:32: error: 'x' was not declared in this scope
   11 |     cout<<"x after loop is: "<<x<<endl;
      |                                ^
D:/UET-LHR/Spring-26/OOP-BS/Test Codes/Test/main.cpp:12:32: error: 'y' was not declared in this scope
   12 |     cout<<"y after loop is: "<<y<<endl;
      |                                ^
ninja: build stopped: subcommand failed.
```

For Loop



9. A variable defined within a block is visible

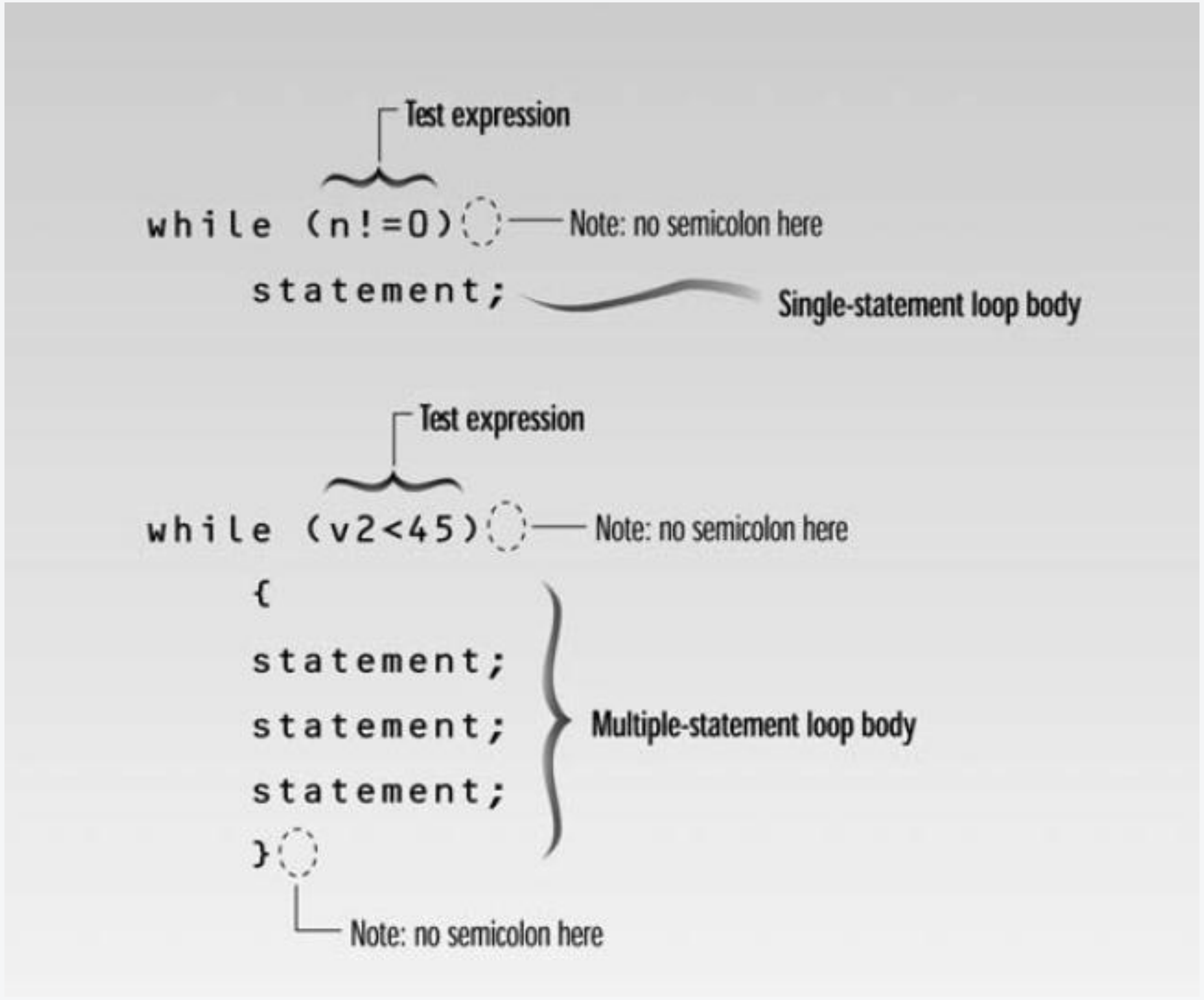
   a. from the point of definition onward in the program.

   b. from the point of definition onward in the function.

   c. from the point of definition onward in the block.

   d. throughout the function.

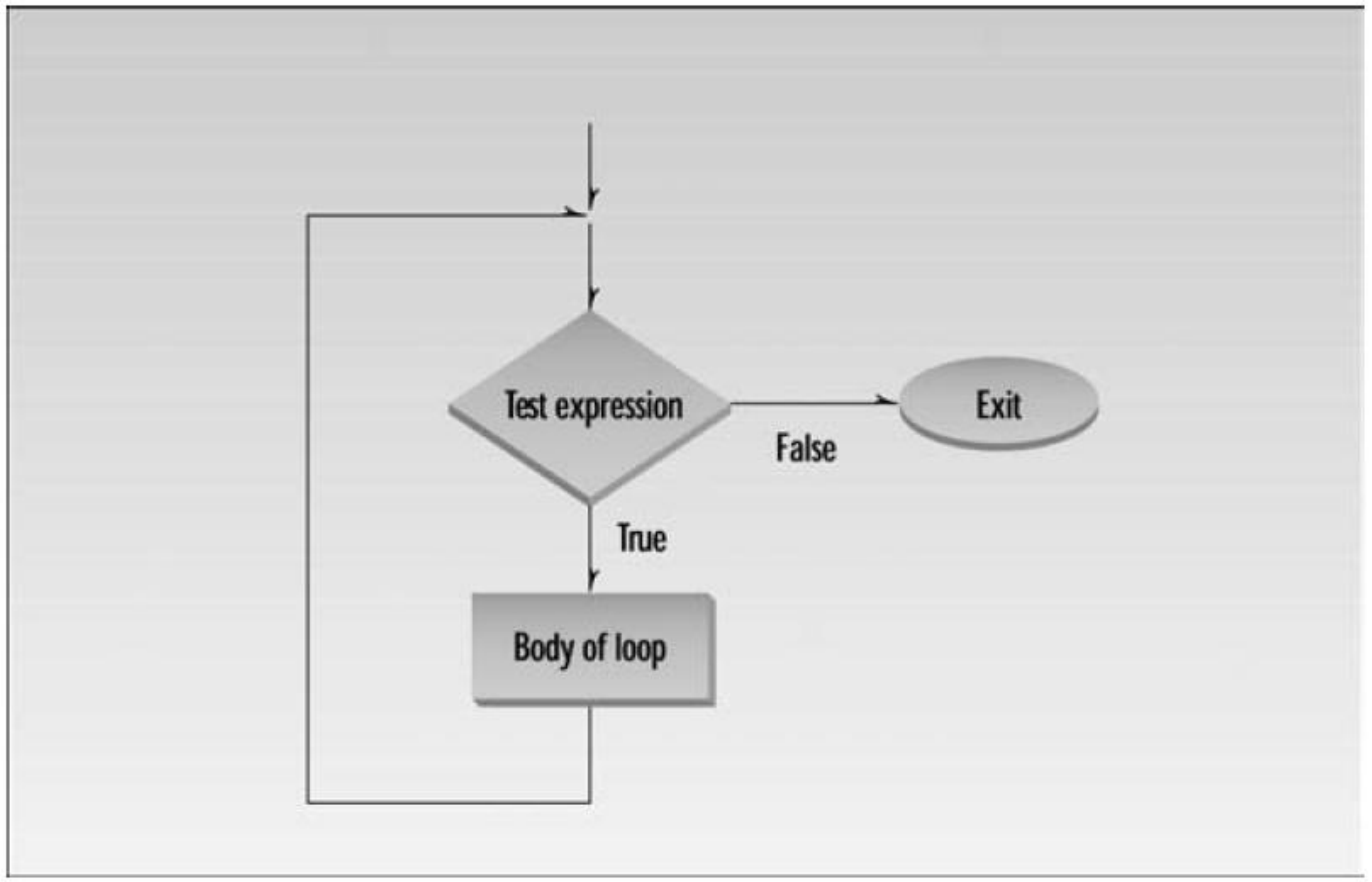
While Loop



```
while (n!=0)
    statement;
```

```
while (v2<45)
    {
    statement;
    statement;
    statement;
    }
```

While Loop

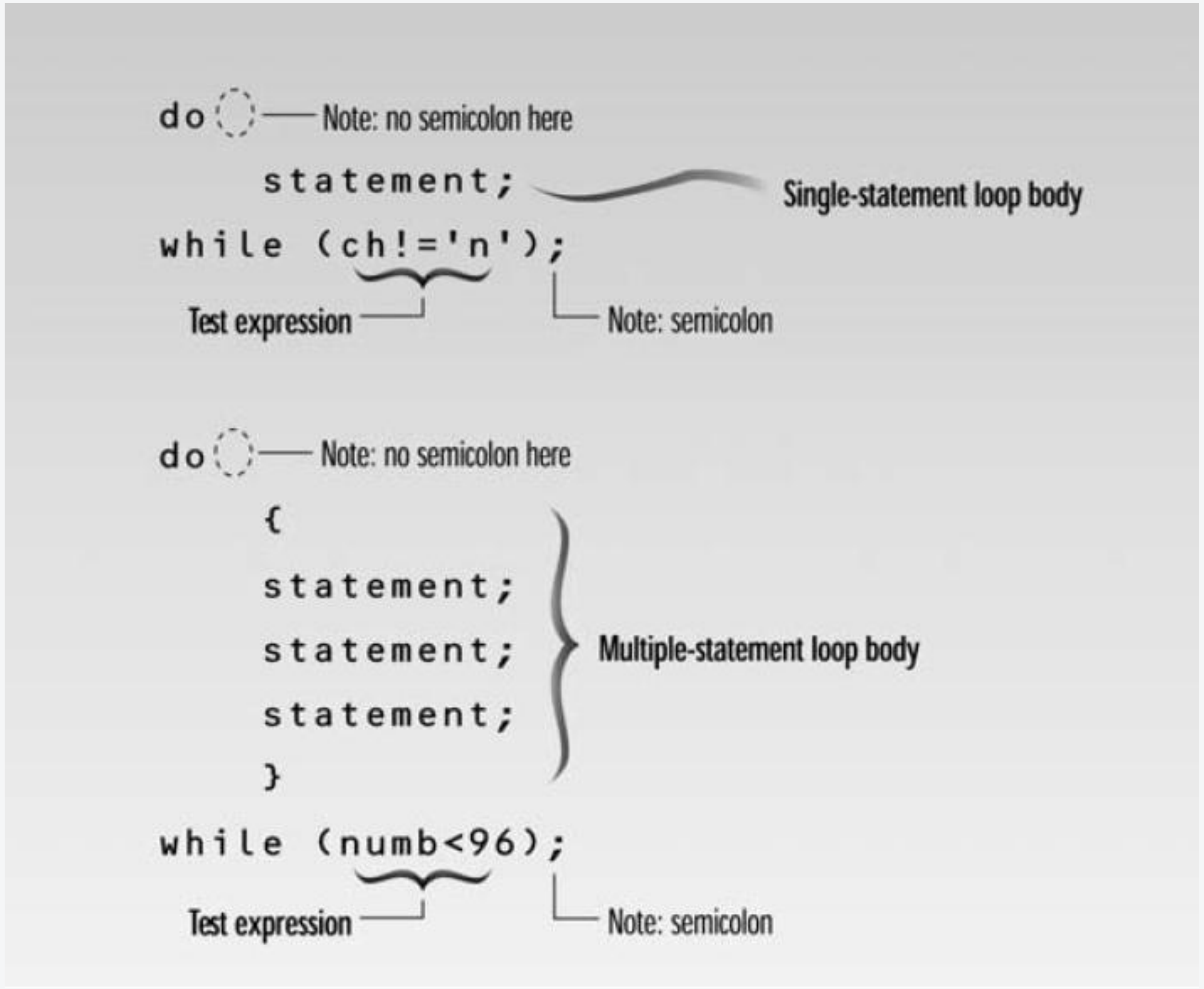


While Loop

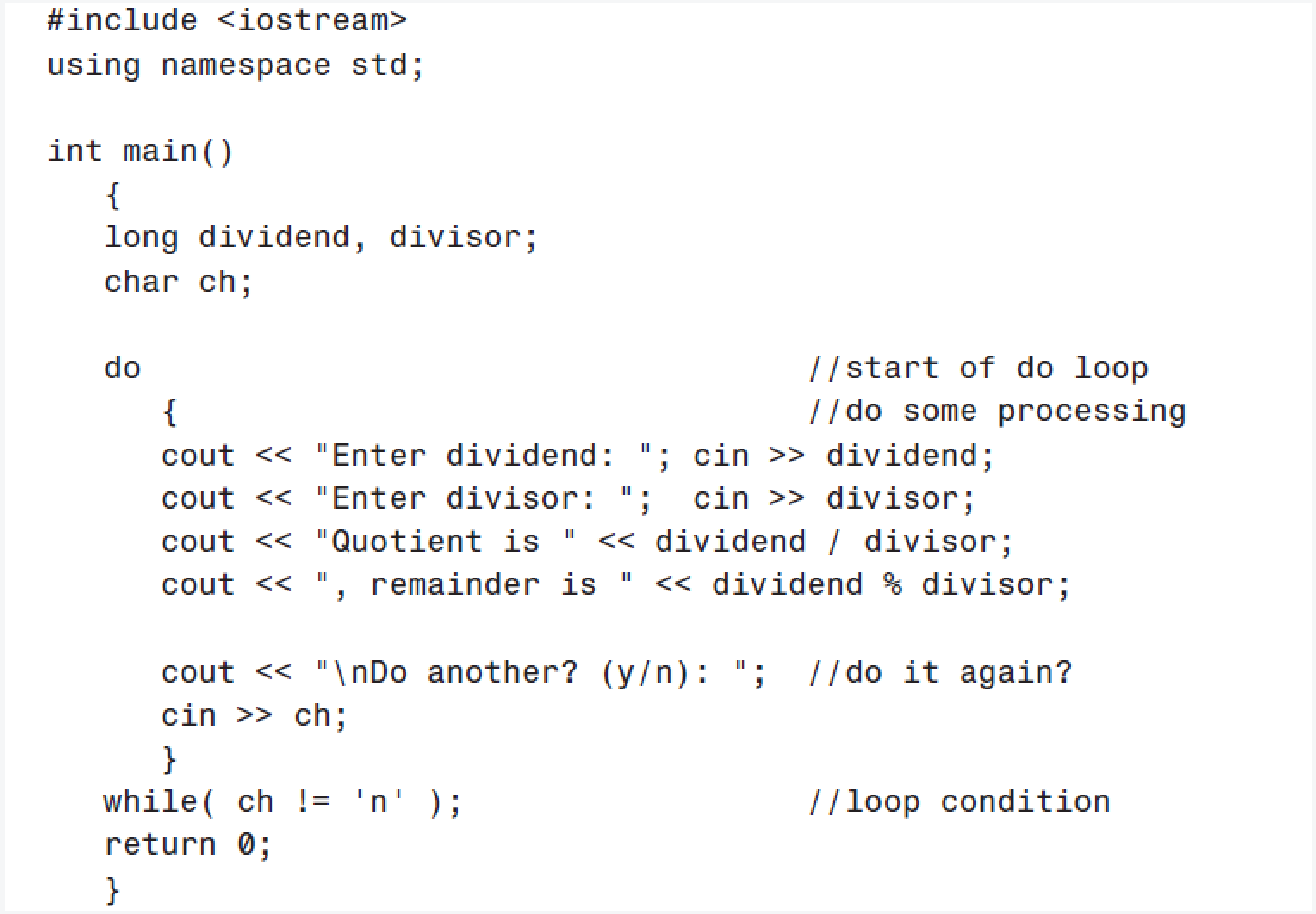
```
#include <iostream>
#include <iomanip>                  //for setw
using namespace std;

int main()
   {
   int pow=1;                       //power initially 1
   int numb=1;                      //numb goes from 1 to ???

   while( pow<10000 )               //loop while power <= 4 digits
      {
      cout << setw(2) << numb;         //display number
      cout << setw(5) << pow << endl;  //display fourth power
      ++numb;                          //get ready for next power
      pow = numb*numb*numb*numb;       //calculate fourth power
      }
   cout << endl;
   return 0;
   }
```

```
 1    1
 2   16
 3   81
 4  256
 5  625
 6 1296
 7 2401
 8 4096
 9 6561
```

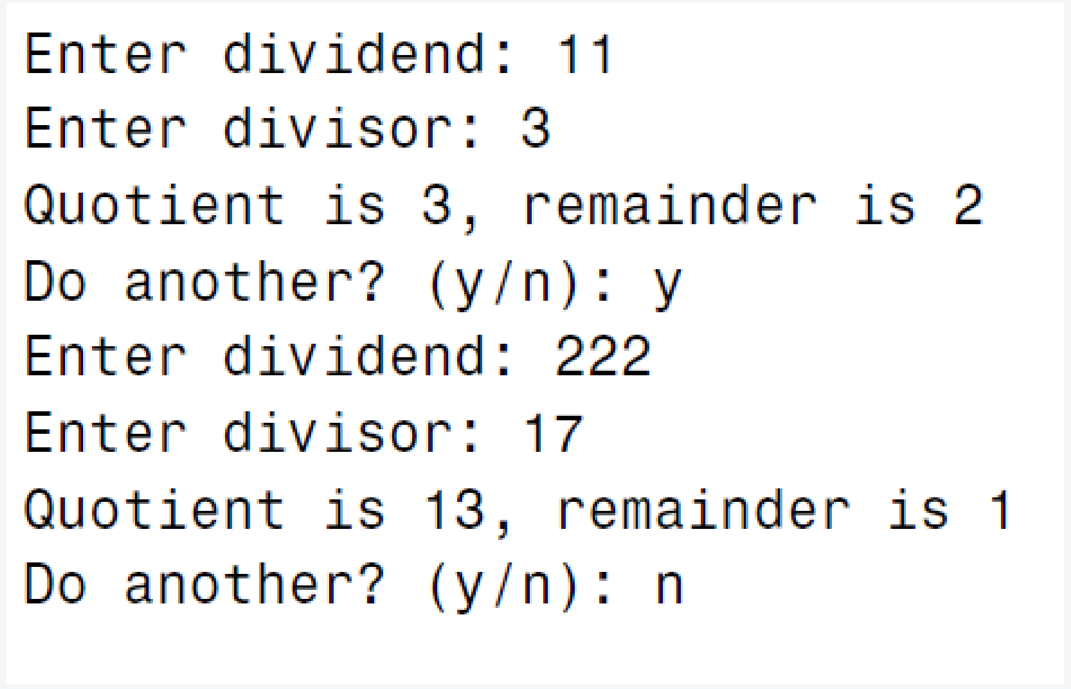
do Loop


```
do
    statement;
while (ch!='n');

do
    {
    statement;
    statement;
    statement;
    }
while (numb<96);
```

do Loop



```
#include <iostream>
using namespace std;

int main()
   {
   long dividend, divisor;
   char ch;

   do                                  //start of do loop
      {                                //do some processing
      cout << "Enter dividend: "; cin >> dividend;
      cout << "Enter divisor: ";  cin >> divisor;
      cout << "Quotient is " << dividend / divisor;
      cout << ", remainder is " << dividend % divisor;

      cout << "\nDo another? (y/n): ";  //do it again?
      cin >> ch;
      }
   while( ch != 'n' );                 //loop condition
   return 0;
   }
```

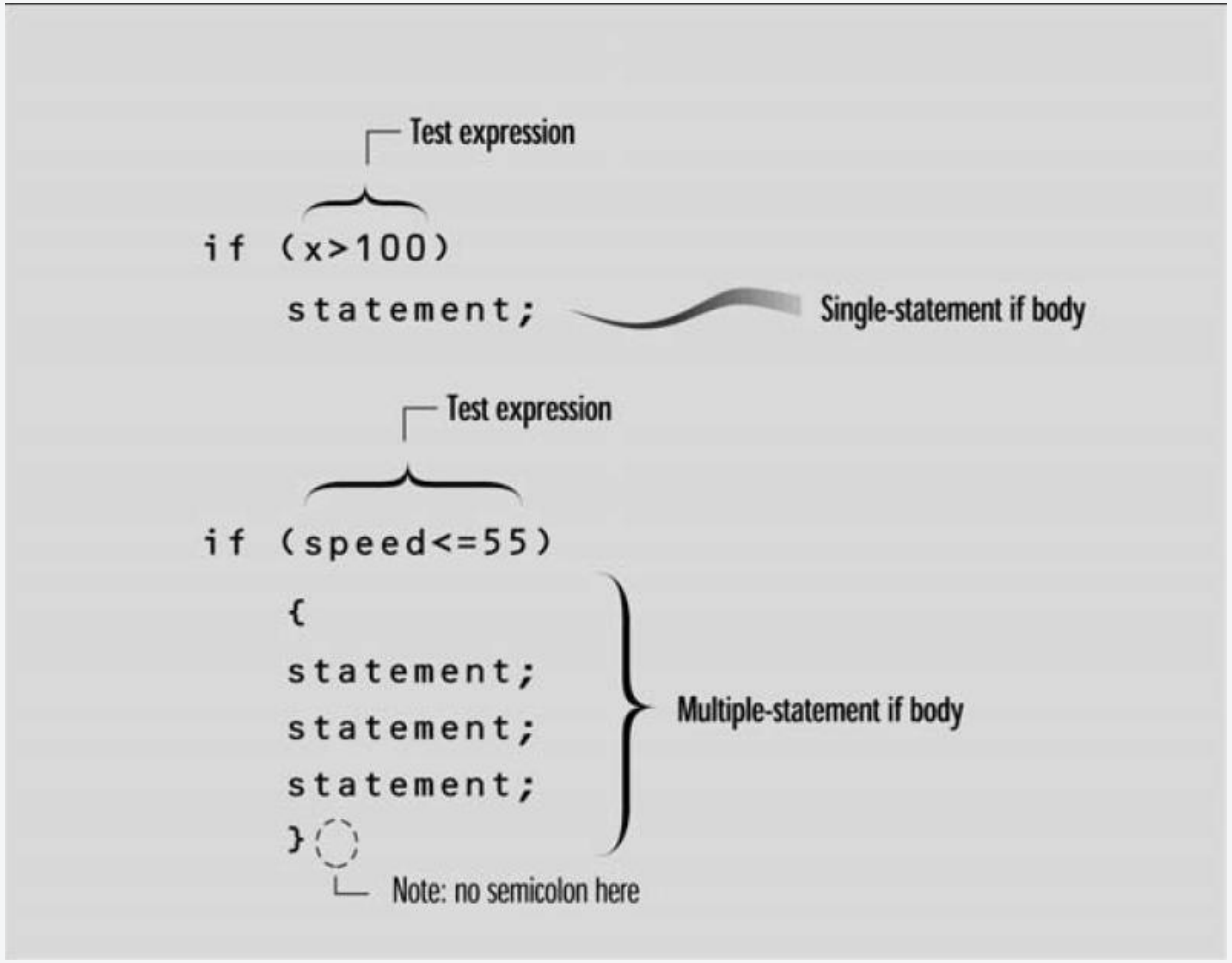


```
Enter dividend: 11
Enter divisor: 3
Quotient is 3, remainder is 2
Do another? (y/n): y
Enter dividend: 222
Enter divisor: 17
Quotient is 13, remainder is 1
Do another? (y/n): n
```

03.

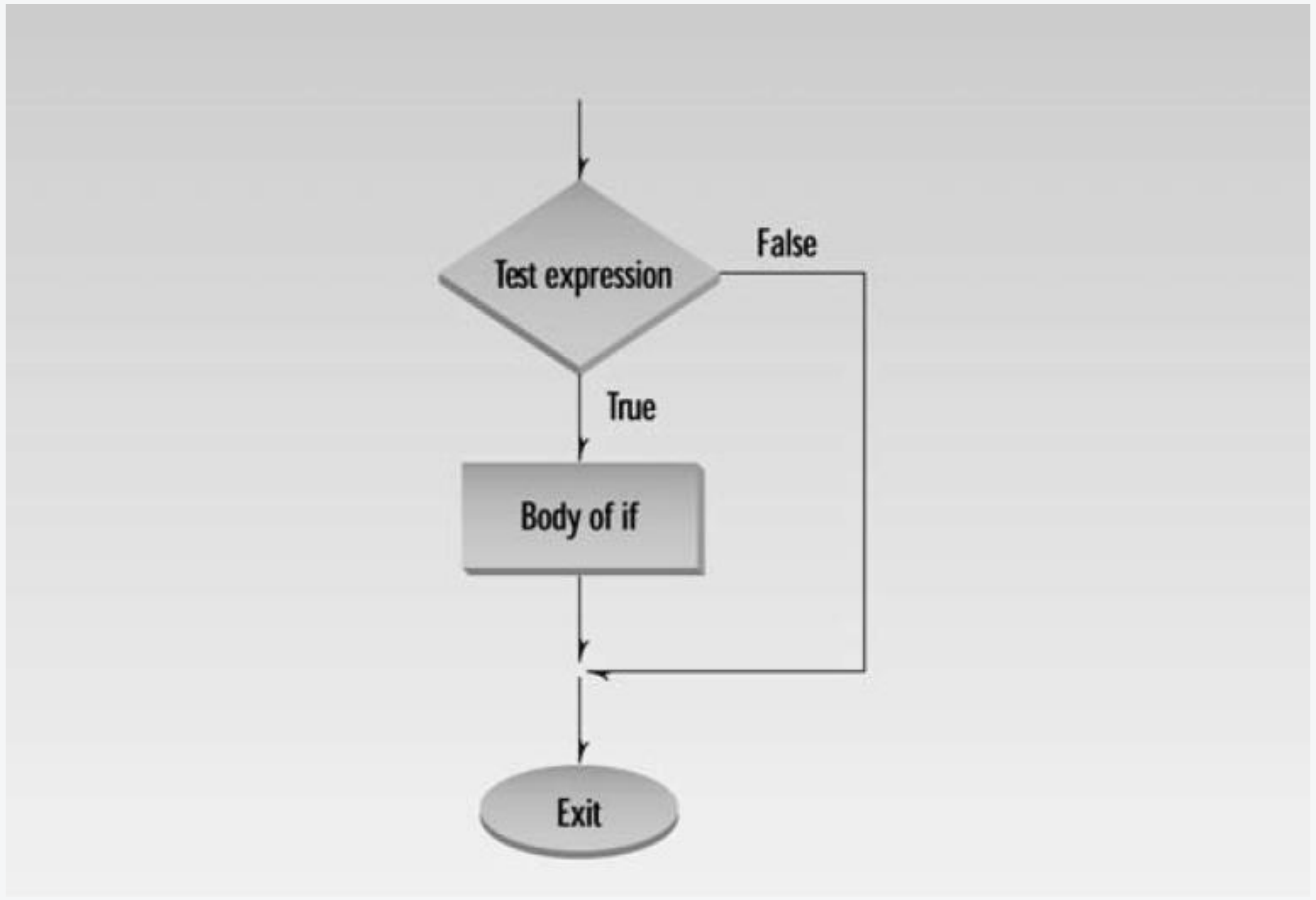
Decision/ Conditionals

If statement



```
if (x>100)
    statement;

if (speed<=55)
    {
    statement;
    statement;
    statement;
    }
```

If statement



If statement

```
#include <iostream>
using namespace std;

int main()
   {
   int x;

   cout << "Enter a number: ";
   cin >> x;
   if( x > 100 )
      cout << "That number is greater than 100\n";
   return 0;
   }
```

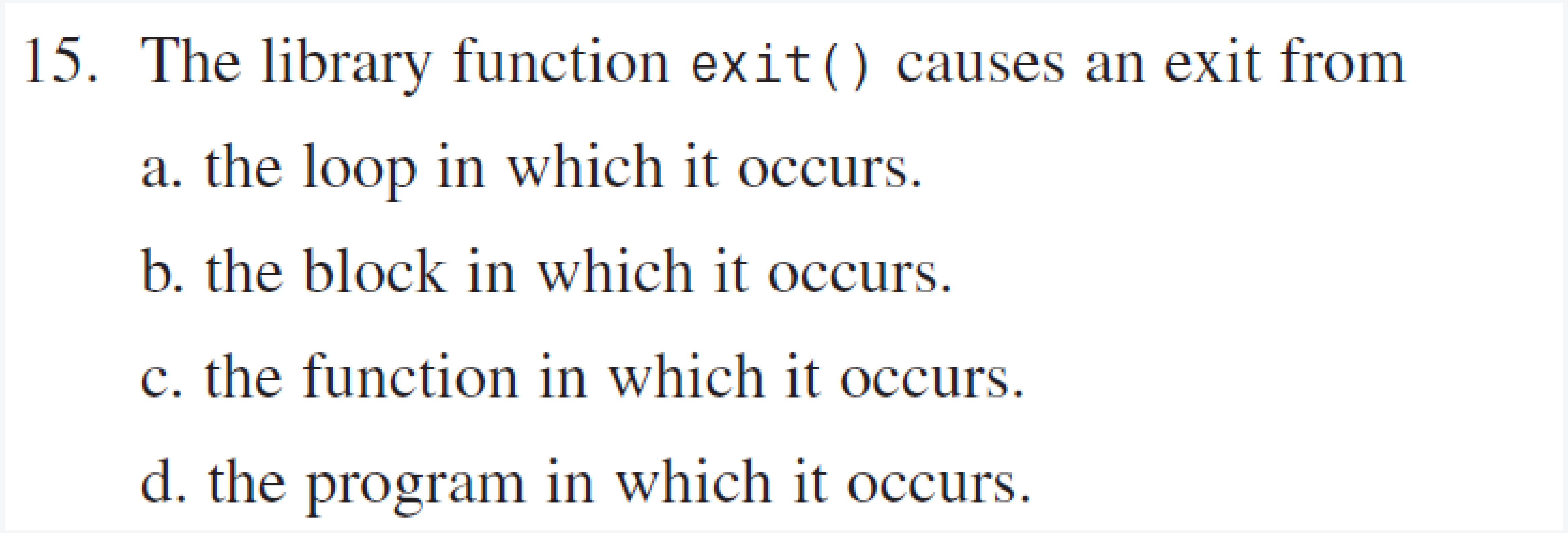
```
Enter a number: 2000
That number is greater than 100
```

If statement

```
#include <iostream>
using namespace std;
#include <process.h>                 //for exit()

int main()
   {
   unsigned long n, j;
  cout << "Enter a number: ";
  cin >> n;                         //get number to test
  for(j=2; j <= n/2; j++)           //divide by every integer from
     if(n%j == 0)                   //2 on up; if remainder is 0,
        {                           //it's divisible by j
        cout << "It's not prime; divisible by " << j << endl;
        exit(0);                    //exit from the program
        }
  cout << "It's prime\n";
  return 0;
  }
```

```
Enter a number: 13
It's prime
Enter a number: 22229
It's prime
Enter a number: 22231
It's not prime; divisible by 11
```

If statement



15. The library function `exit()` causes an exit from

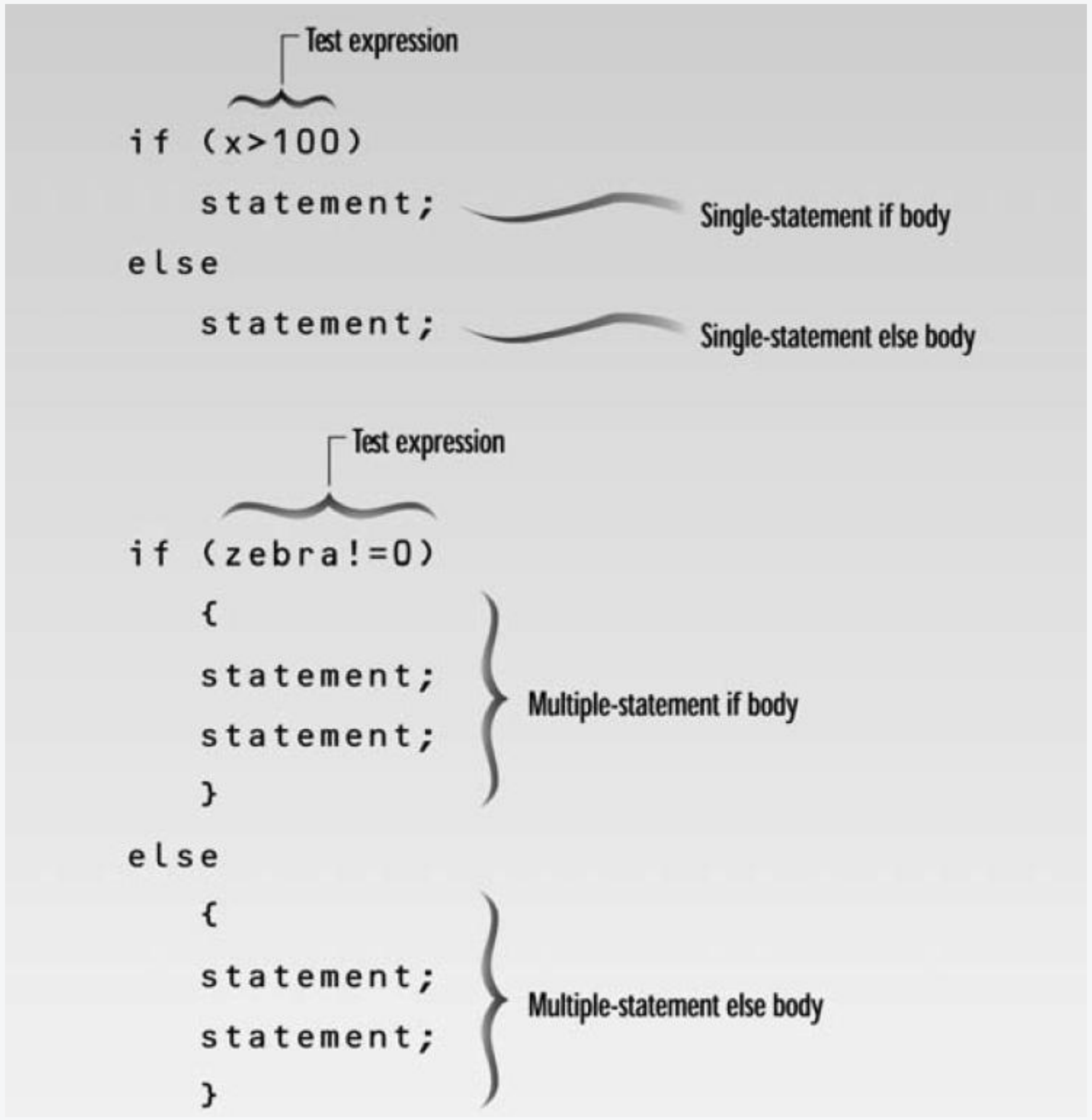
    a. the loop in which it occurs.

    b. the block in which it occurs.

    c. the function in which it occurs.

    d. the program in which it occurs.

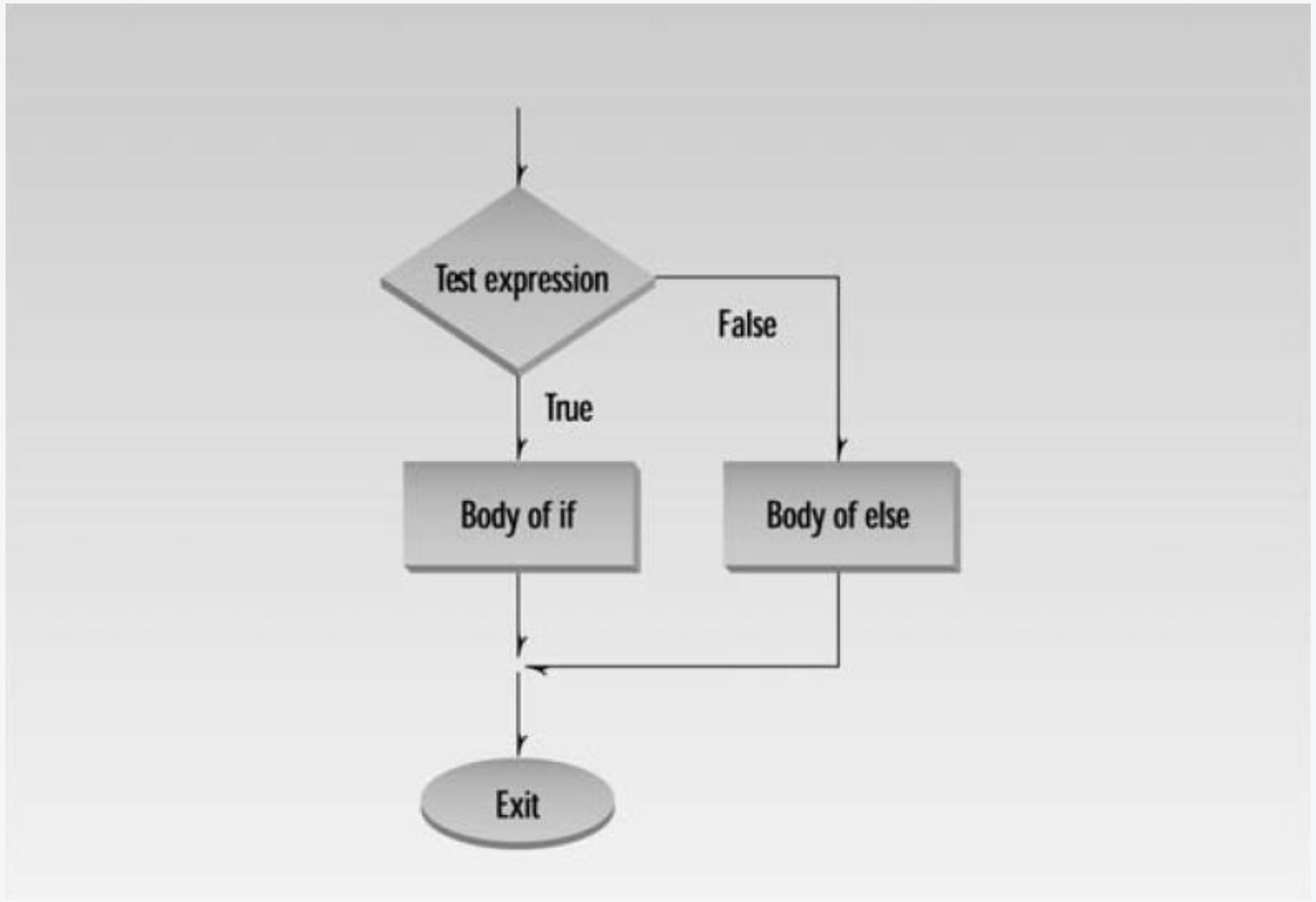
If-else statement



```
if (x>100)
   statement;
else
   statement;

if (zebra!=0)
   {
   statement;
   statement;
   }
else
   {
   statement;
   statement;
   }
```

Test expression
Single-statement if body
Single-statement else body
Test expression
Multiple-statement if body
Multiple-statement else body

If-else statement



If-else statement

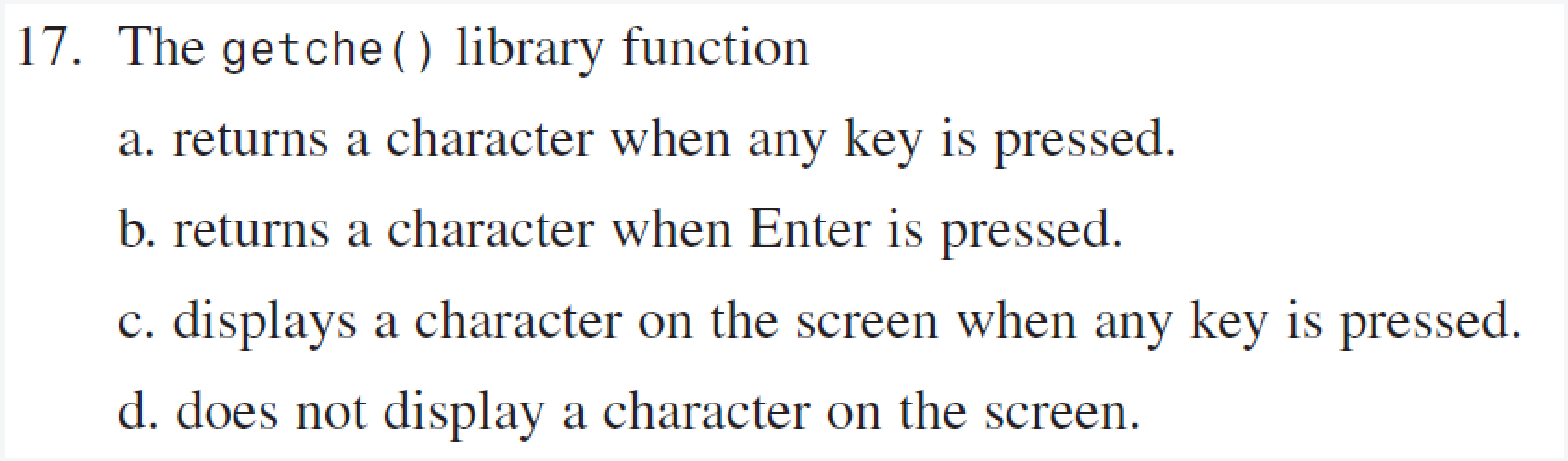
```
int main()
   {
   int x;

   cout << "\nEnter a number: ";
   cin >> x;
   if( x > 100 )
      cout << "That number is greater than 100\n";
   else
      cout << "That number is not greater than 100\n";
   return 0;
   }
```

If-else statement

```
#include <iostream>
using namespace std;
#include <conio.h>              //for getche()

int main()
   {
   int chcount=0;               //counts non-space characters
   int wdcount=1;               //counts spaces between words
   char ch = 'a';               //ensure it isn't '\r'

   cout << "Enter a phrase: ";
   while( ch != '\r' )          //loop until Enter typed
      {
      ch = getche();            //read one character
      if( ch==' ' )             //if it's a space
      wdcount++;                //count a word
      else                      //otherwise,
      chcount++;                //count a character
      }                         //display results
   cout << "\nWords=" << wdcount << endl
        << "Letters=" << (chcount-1) << endl;
   return 0;
   }
```

```
For while and do
Words=4
Letters=13
```

If-else statement



17. The `getche()` library function

    a. returns a character when any key is pressed.

    b. returns a character when Enter is pressed.

    c. displays a character on the screen when any key is pressed.

    d. does not display a character on the screen.

If-else statement

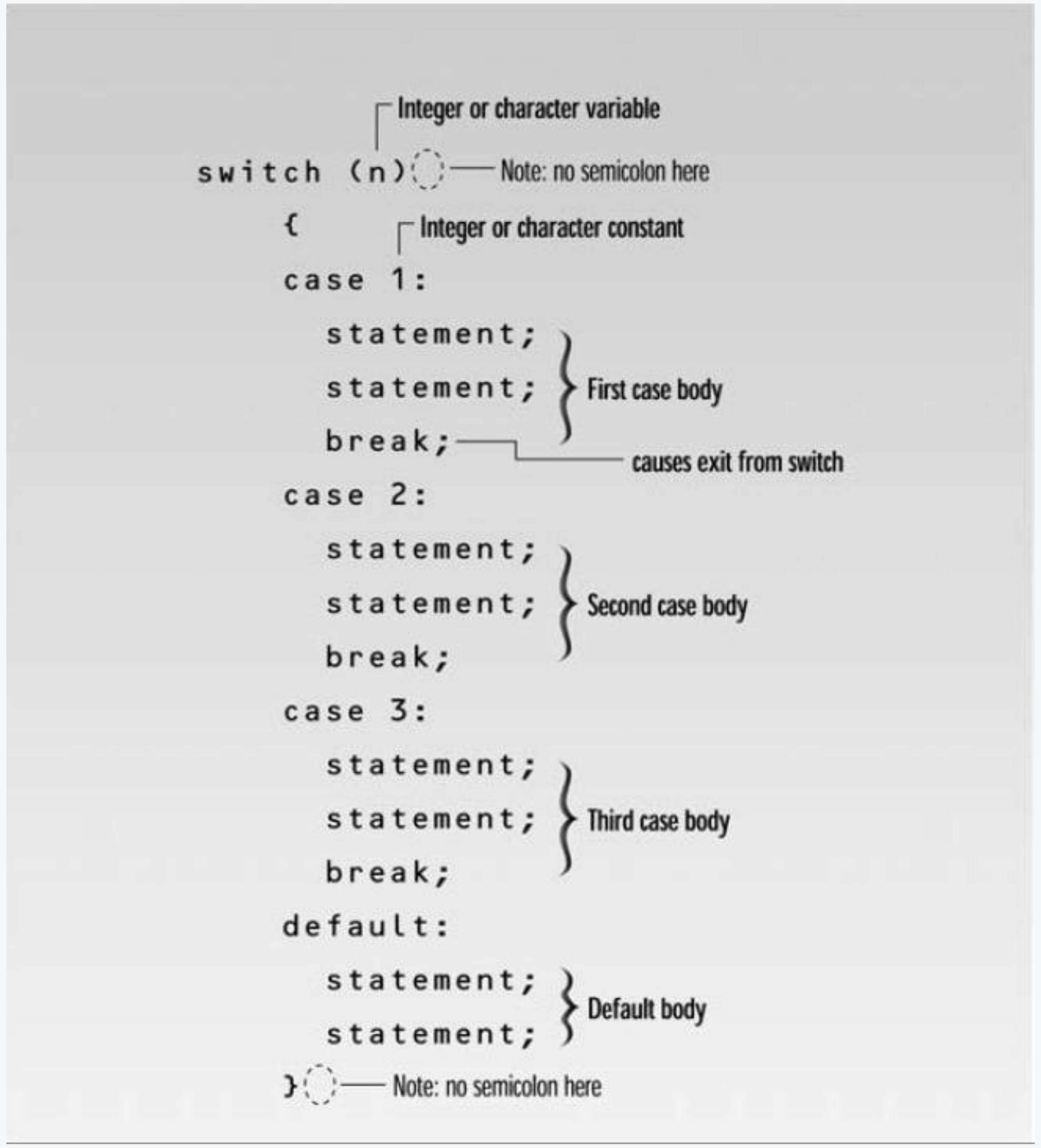
18. What is the character obtained from `cin` when the user presses the Enter key?

If-else if statement

```
#include <iostream>
using namespace std;
#include <conio.h>                  //for getche()

int main()
   {
   char dir='a';
   int x=10, y=10;

   cout << "Type Enter to quit\n";
   while( dir != '\r' )             //until Enter is typed
      {
      cout << "\nYour location is " << x << ", " << y;
      cout << "\nPress direction key (n, s, e, w): ";
      dir = getche();               //get character
      if( dir=='n')                 //go north
         y--;
      else if( dir=='s' )           //go south
         y++;
      else if( dir=='e' )           //go east
         x++;
      else if( dir=='w' )           //go west
         x--;
      }  //end while
   return 0;
   }  //end main
```

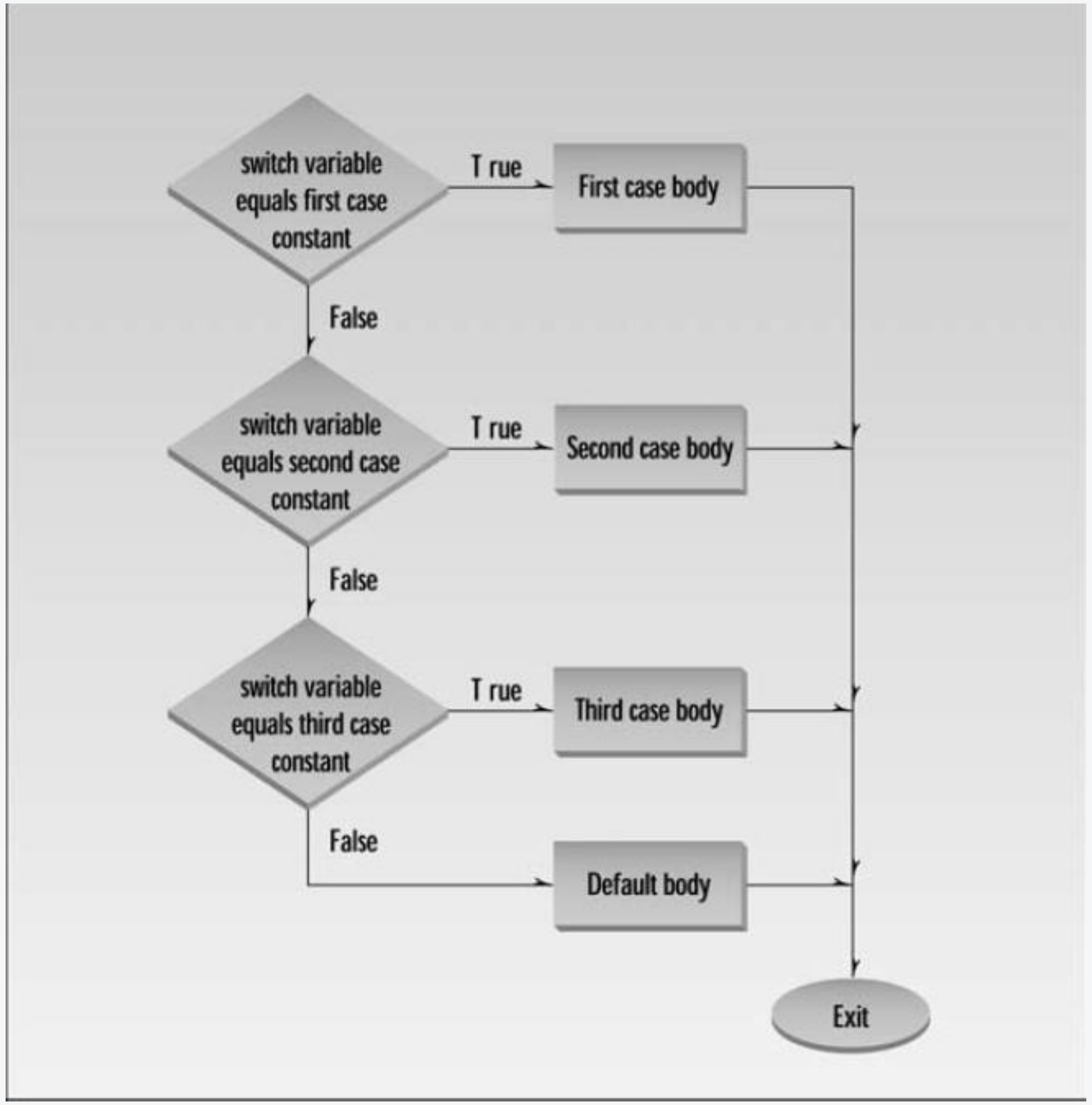
```
Your location is 10, 10
Press direction key (n, s, e, w): n
Your location is 10, 9
Press direction key (n, s, e, w): e
Your location is 11, 9
Press direction key (n, s, e, w):
```

Switch statement

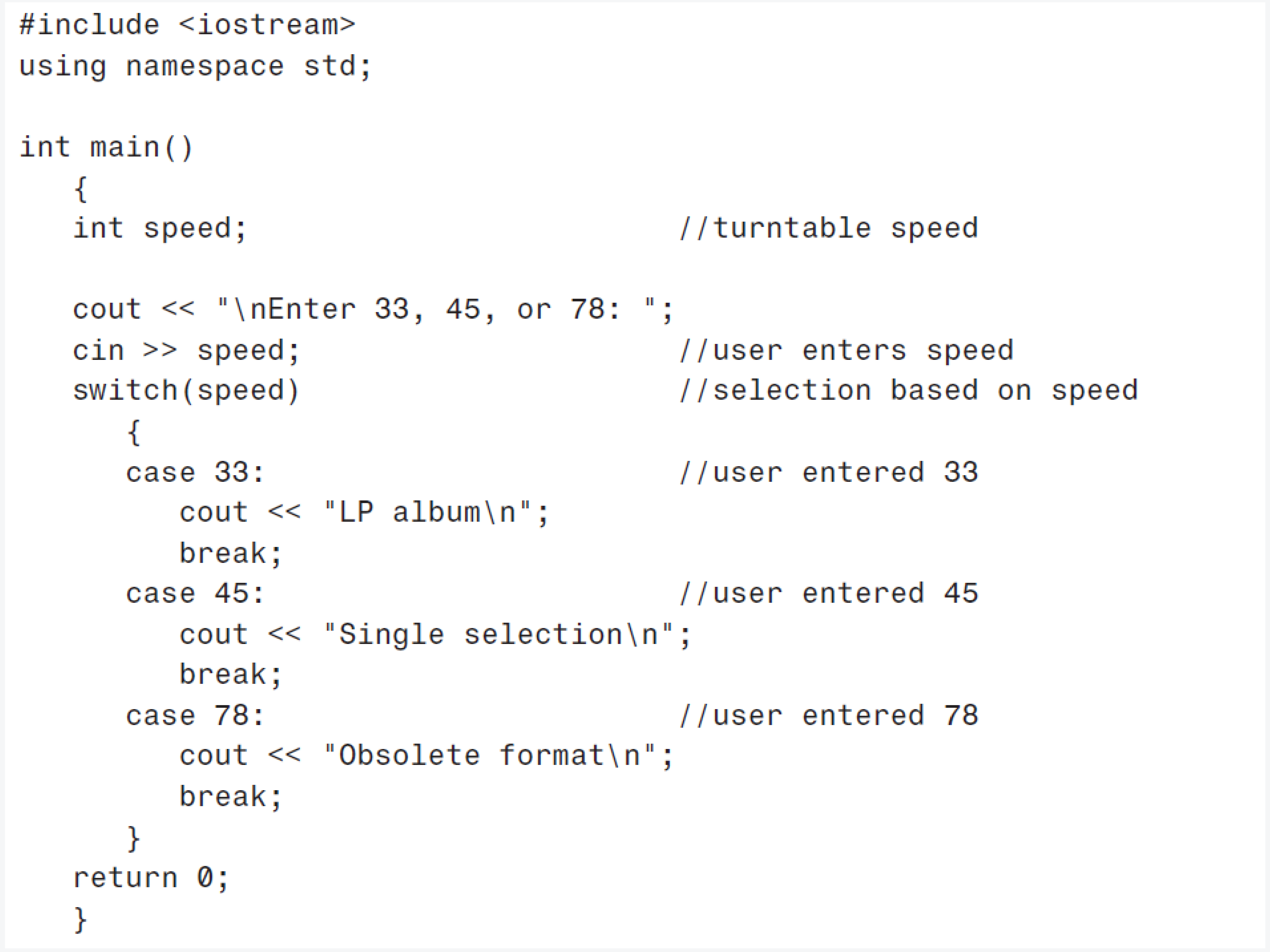


```
switch (n)
  {
  case 1:
     statement;
     statement;
     break;
  case 2:
     statement;
     statement;
     break;
  case 3:
     statement;
     statement;
     break;
  default:
     statement;
     statement;
  }
```

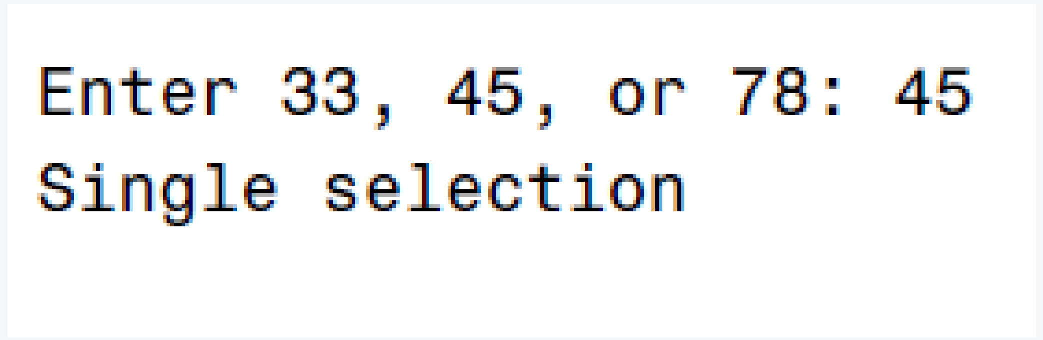
Switch statement


switch variable equals first case constant
T rue
First case body
False
switch variable equals second case constant
T rue
Second case body
False
switch variable equals third case constant
T rue
Third case body
False
Default body
Exit

Switch statement



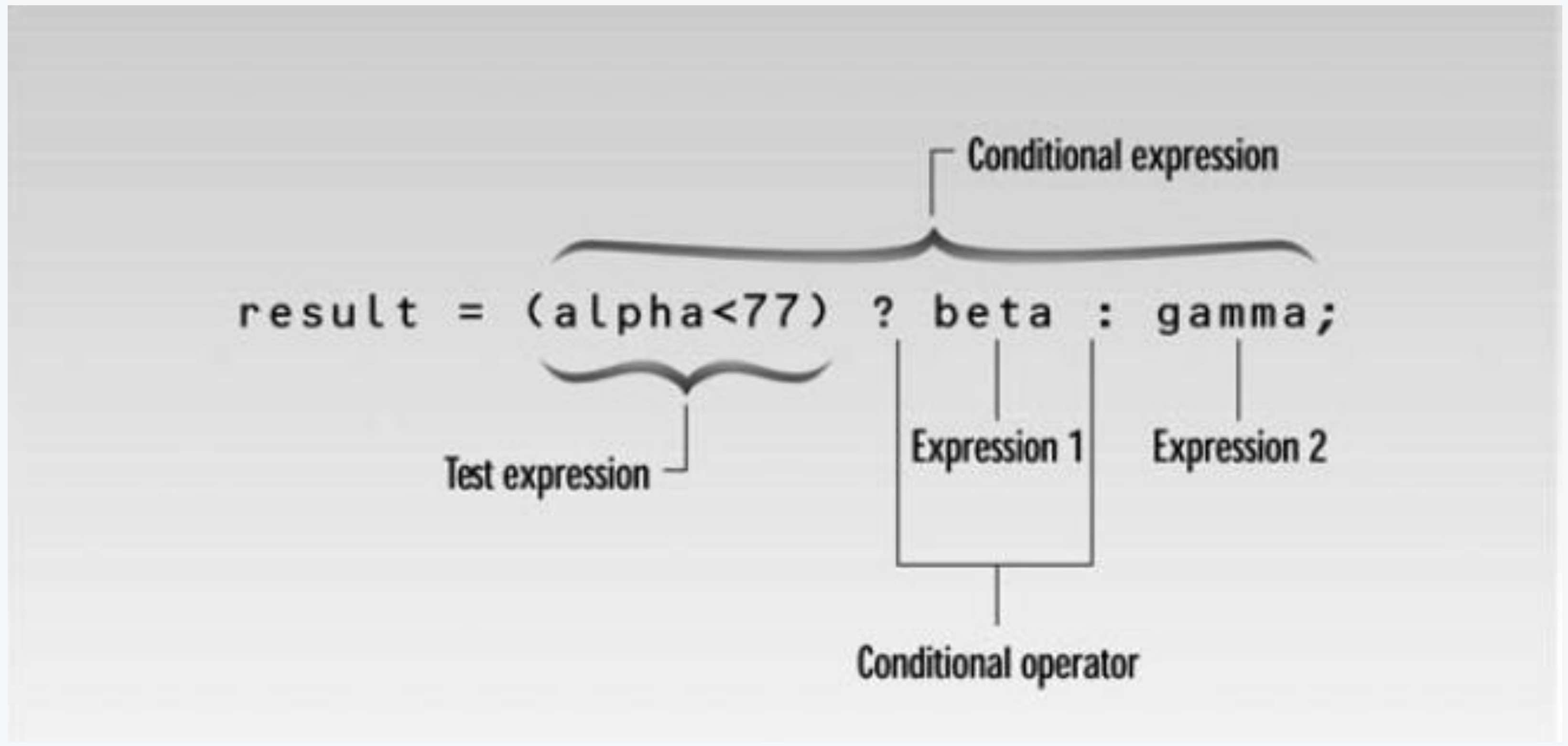
```
#include <iostream>
using namespace std;

int main()
   {
   int speed;                         //turntable speed

   cout << "\nEnter 33, 45, or 78: ";
   cin >> speed;                      //user enters speed
   switch(speed)                      //selection based on speed
      {
      case 33:                        //user entered 33
         cout << "LP album\n";
         break;
      case 45:                        //user entered 45
         cout << "Single selection\n";
         break;
      case 78:                        //user entered 78
         cout << "Obsolete format\n";
         break;
      }
   return 0;
   }
```



```
Enter 33, 45, or 78: 45
Single selection
```

Conditional Operator

```
if( alpha < beta )
   min = alpha;
else
   min = beta;
```



```
min = (alpha<beta) ? alpha : beta;
```

04.

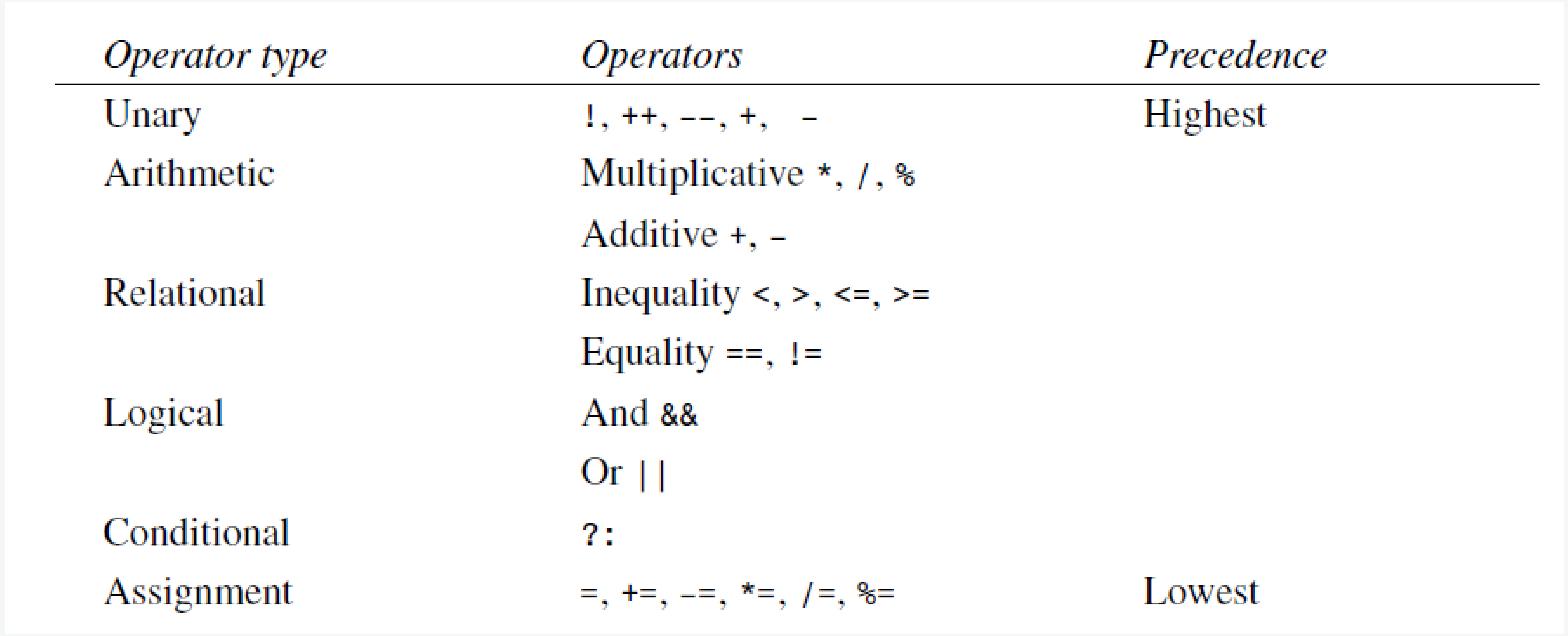
Logical Operators

Logical Operators

Operator	*Effect*
&&	Logical AND
||	Logical OR
!	Logical NOT

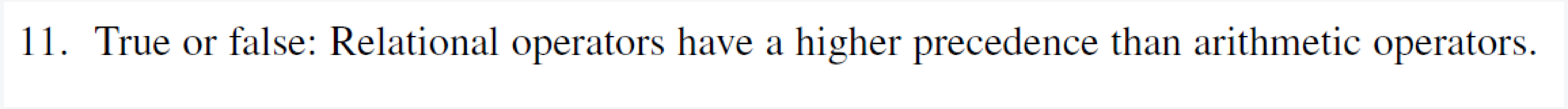
```
#include <iostream>
using namespace std;
#include <process.h>                //for exit()
#include <conio.h>                  //for getche()

int main()
   {
   char dir='a';
   int x=10, y=10;

   while( dir != '\r' )
      {
      cout << "\nYour location is " << x << ", " << y;
      cout << "\nEnter direction (n, s, e, w): ";
      dir = getche();               //get direction
      switch(dir)
         {
         case 'n': y--; break;      //update coordinates
         case 's': y++; break;
         case 'e': x++; break;
         case 'w': x--; break;
         }
      if( x==7 && y==11 )           //if x is 7 and y is 11
         {
         cout << "\nYou found the treasure!\n";
         exit(0);                   //exit from program
         }
      }  //end switch
   return 0;
   }  //end main
```

Precedence



Operator type	*Operators*	*Precedence*
Unary	`!`, `++`, `--`, `+`, `-`	Highest
Arithmetic	Multiplicative `*`, `/`, `%`	
	Additive `+`, `-`	
Relational	Inequality `<`, `>`, `<=`, `>=`	
	Equality `==`, `!=`	
Logical	And `&&`	
	Or `||`	
Conditional	`?:`	
Assignment	`=`, `+=`, `-=`, `*=`, `/=`, `%=`	Lowest

Precedence

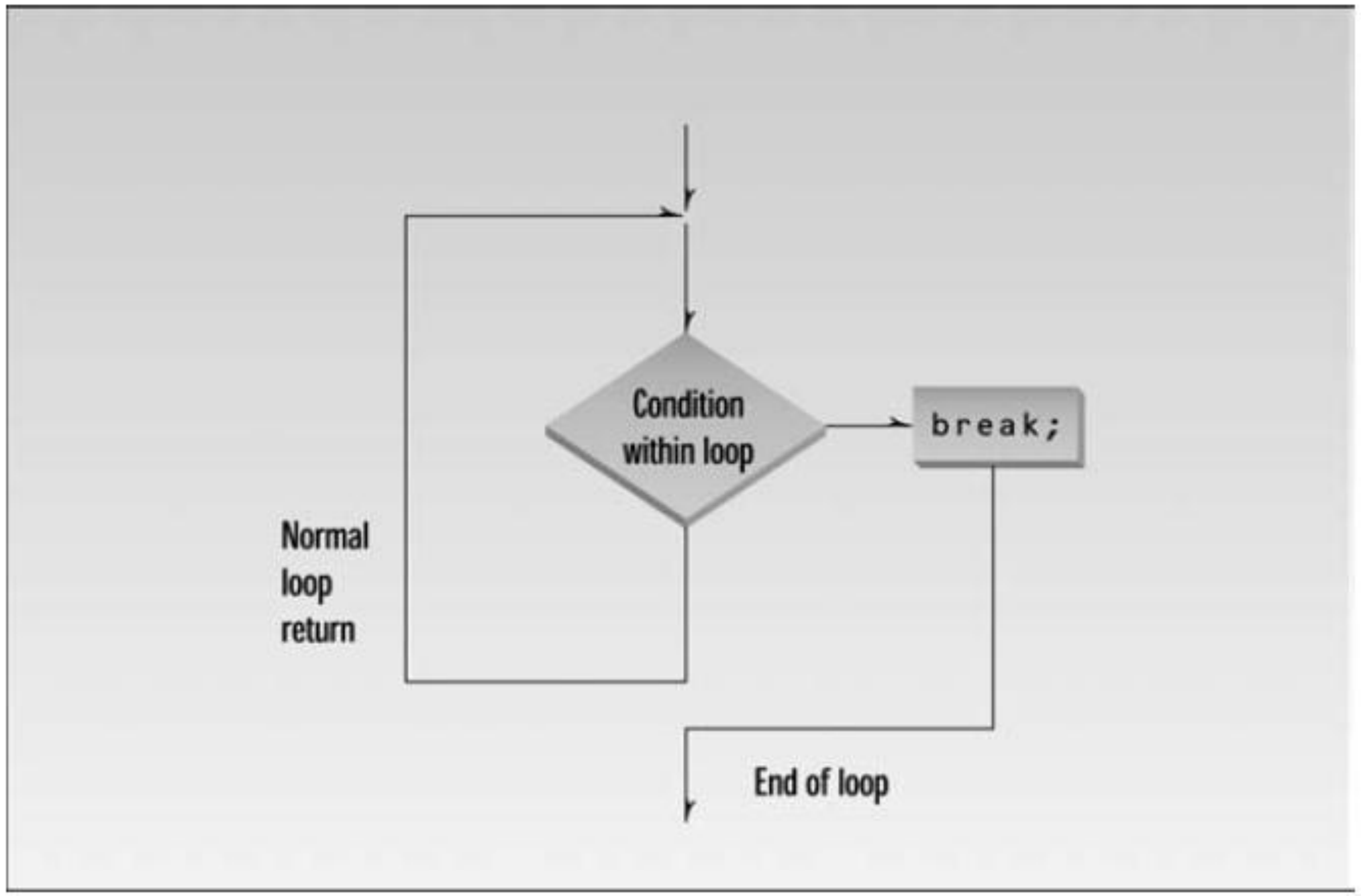


11. True or false: Relational operators have a higher precedence than arithmetic operators.

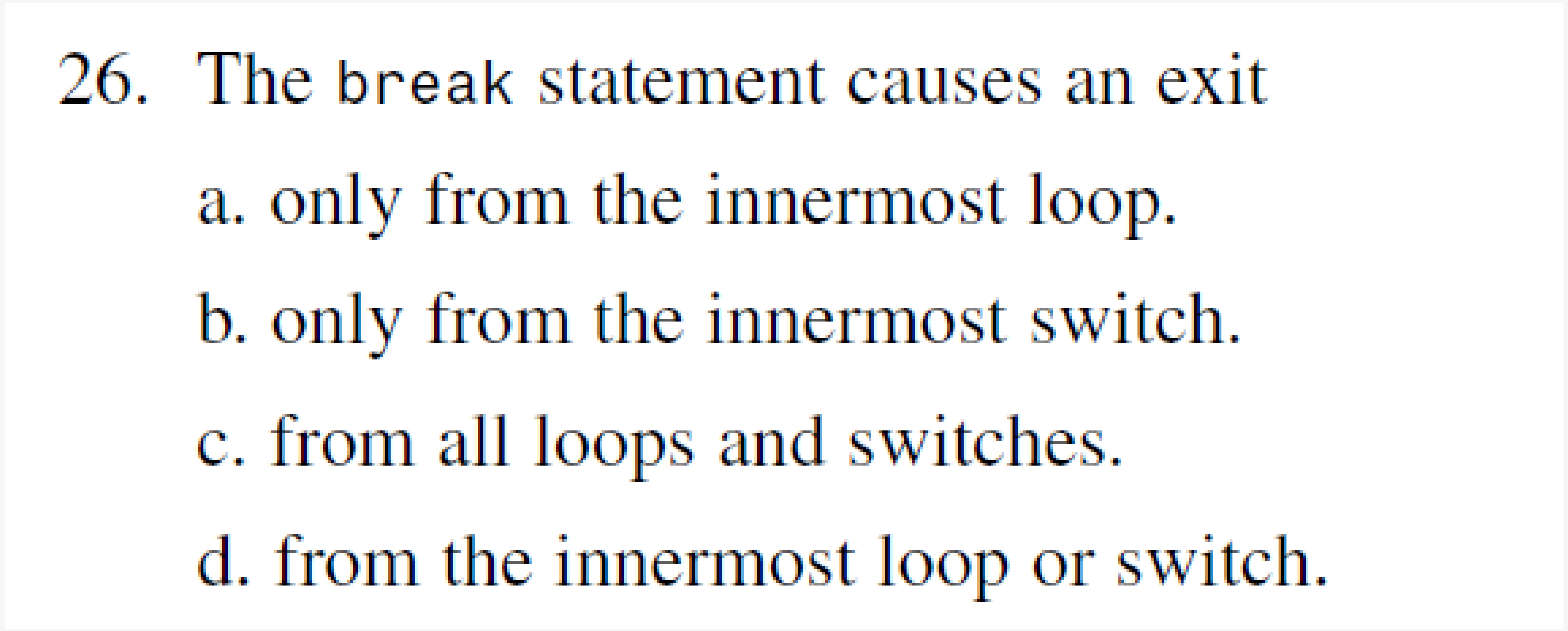
05.

break and continue

break



break



26. The `break` statement causes an exit

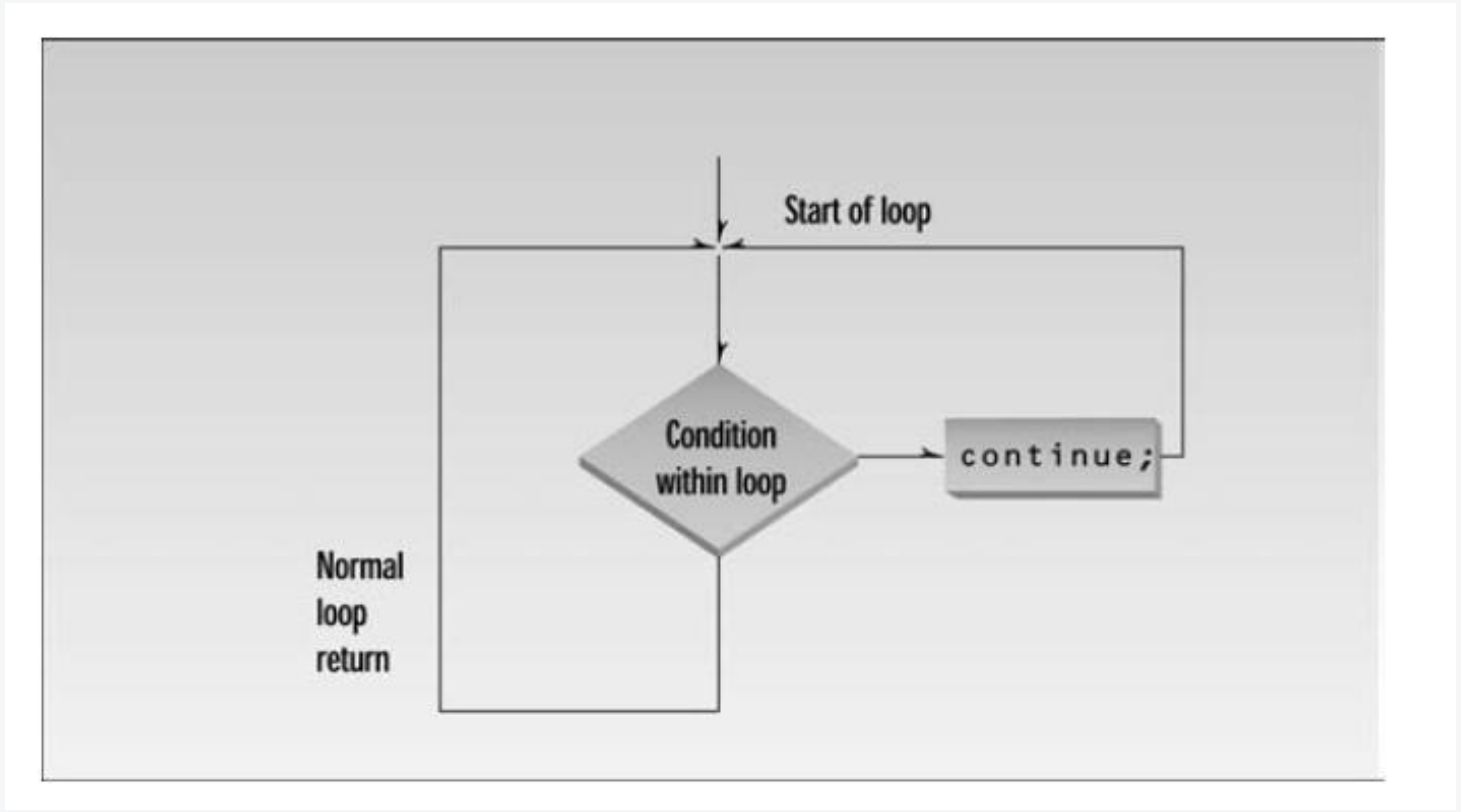
    a. only from the innermost loop.

    b. only from the innermost switch.

    c. from all loops and switches.

    d. from the innermost loop or switch.

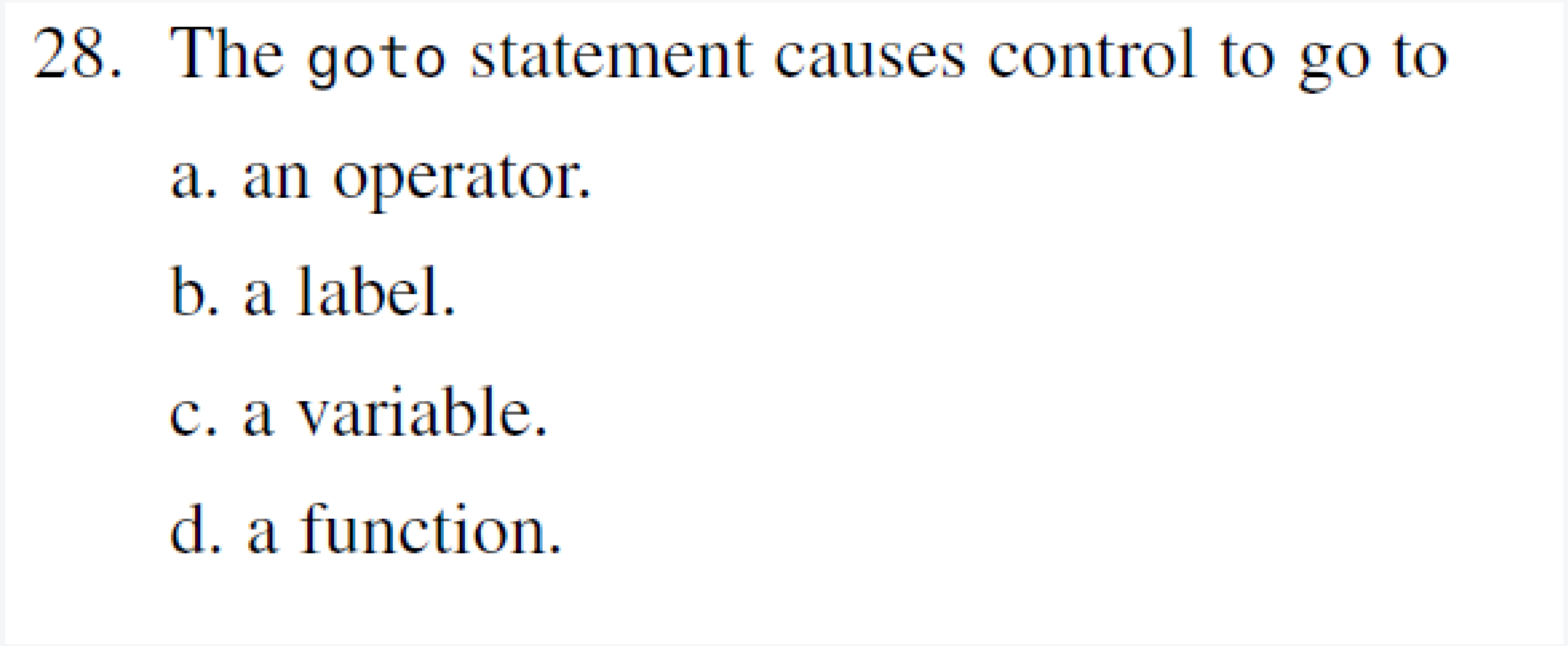
continue


Start of loop
Condition within loop
continue;
Normal loop return

continue

27. Executing the `continue` operator from within a loop causes control to go to ________.

continue



28. The `goto` statement causes control to go to

    a. an operator.

    b. a label.

    c. a variable.

    d. a function.

06.

Tasks

Task-1

- Write a program that allows the user to enter the number and then generates the table, formatting it into 10 columns and 20 lines.
- Interaction with the program should look like this (only the first three lines are shown):
- Enter a number: 7

```
7 14 21 28 35 42 49 56 63 70
77 84 91 98 105 112 119 126 133 140
147 154 161 168 175 182 189 196 203 210
```

Task-2

- Write a program that calculates how much money you'll end up with if you invest an amount of money at a fixed interest rate, compounded yearly.
- Some interaction with the program might look like this:
  - **Enter initial amount: 3000**
  - **Enter number of years: 10**
  - **Enter interest rate (percent per year): 5.5**
- At the end of 10 years, you will have 5124.43 dollars.
- At the end of the first year, you have 3000 + (3000 * 0.055), which is 3165.
- At the end of the second year you have 3165 + (3165 * 0.055), which is 3339.08. Do this as many times as there are years.

Thank You!