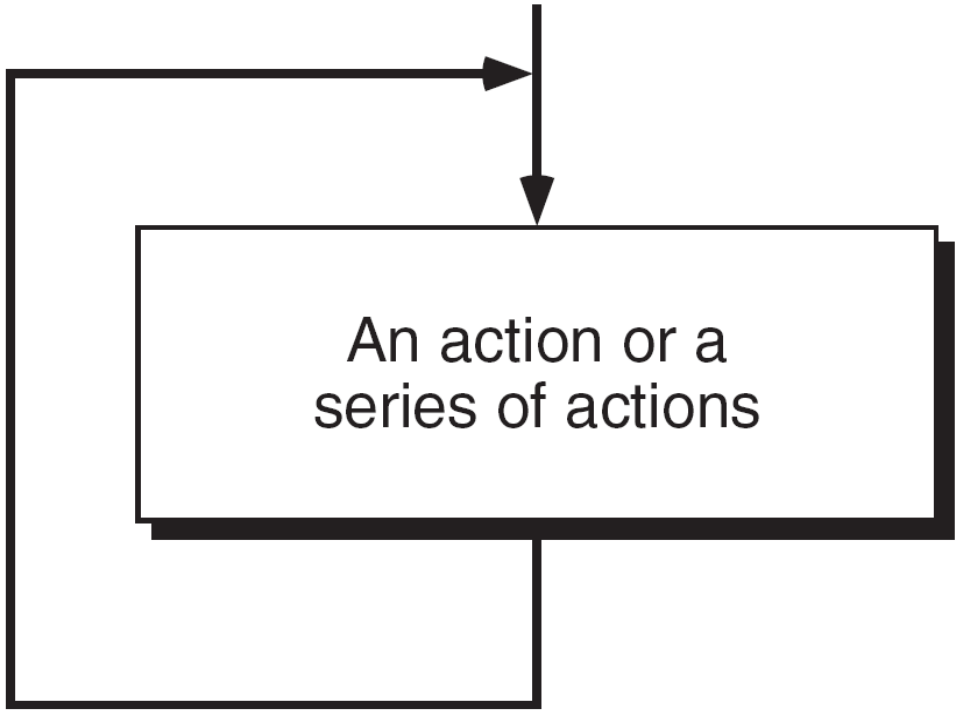
Control Structures in C

Dr Bhanu

6-1 Concept of a loop

The real power of computers is in their ability to repeat an operation or a series of operations many times. This repetition, called looping, is one of the basic structured programming concepts.

Each loop must have an expression that determines if the loop is done. If it is not done, the loop repeats one more time; if it is done, the loop terminates.



A repetition statement (also called an iteration or loop statement) allows you to specify that an action is to be repeated while some condition remains true.

FIGURE 6-1 Concept of a Loop

6-2 Pretest and Post-test Loops

We need to test for the end of a loop, but where should we check it—before or after each iteration? We can have either a pre- or a post-test terminating condition.

In a pretest loop , the condition is checked at the beginning of each iteration.

In a post-test loop, the condition is checked at the end of each iteration.

Note

Pretest Loop

In each iteration, the control expression is tested first. If it is true, the loop continues; otherwise, the loop is terminated.

Post-test Loop

In each iteration, the loop action(s) are executed. Then the control expression is tested. If it is true, a new iteration is started; otherwise, the loop terminates.

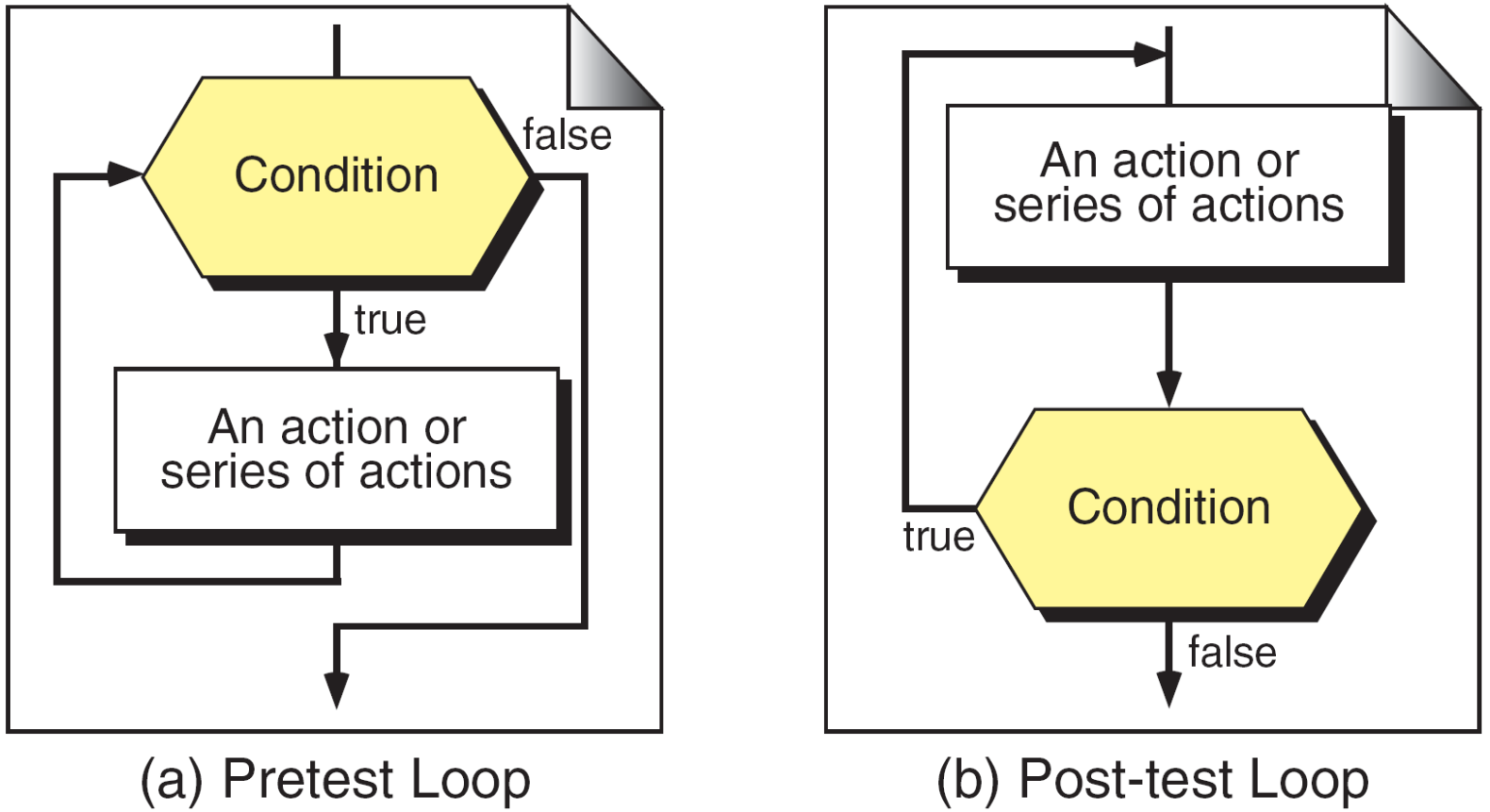


FIGURE 6-2 Pretest and Post-test Loops

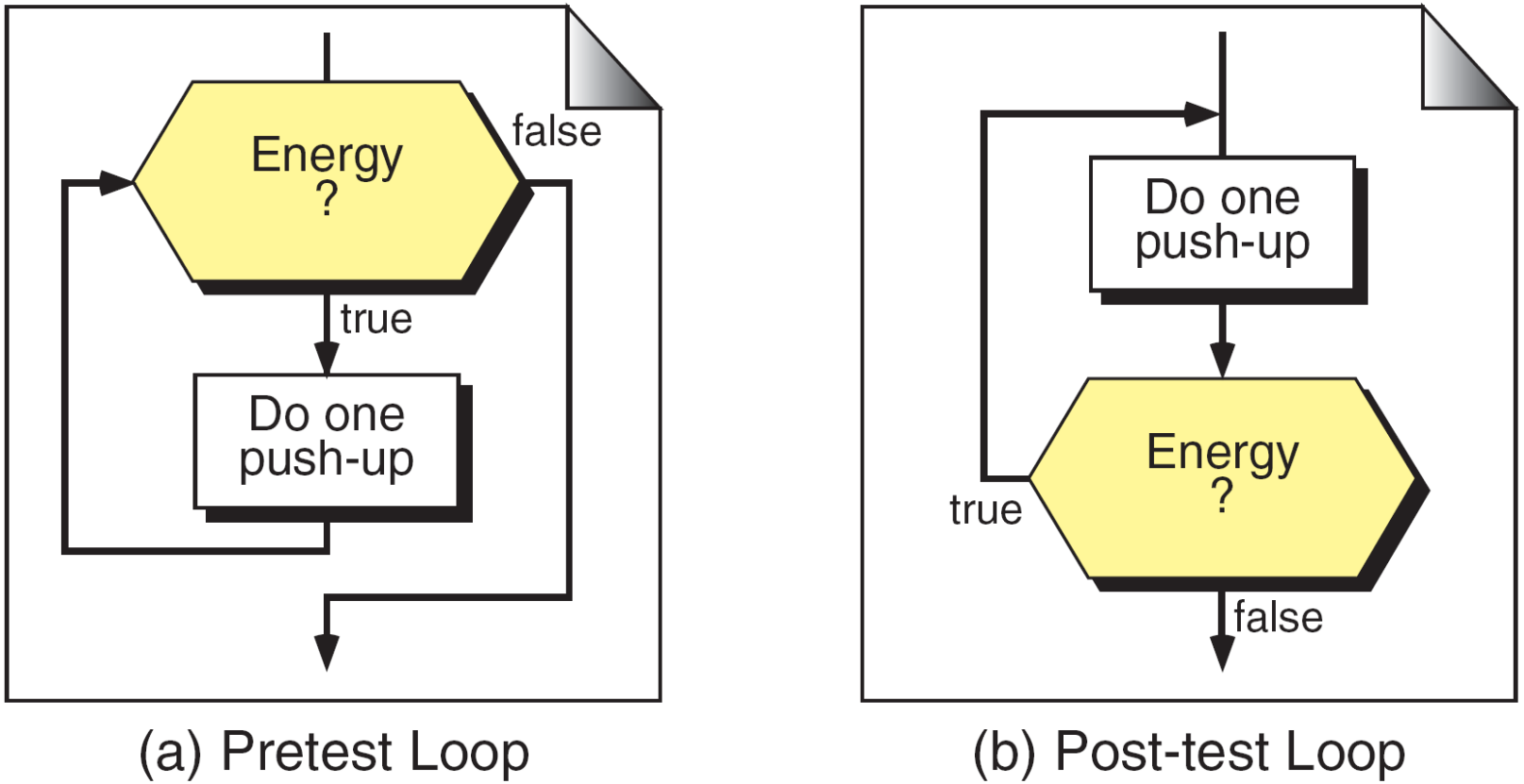


FIGURE 6-3 Two Different Strategies for Doing Exercises

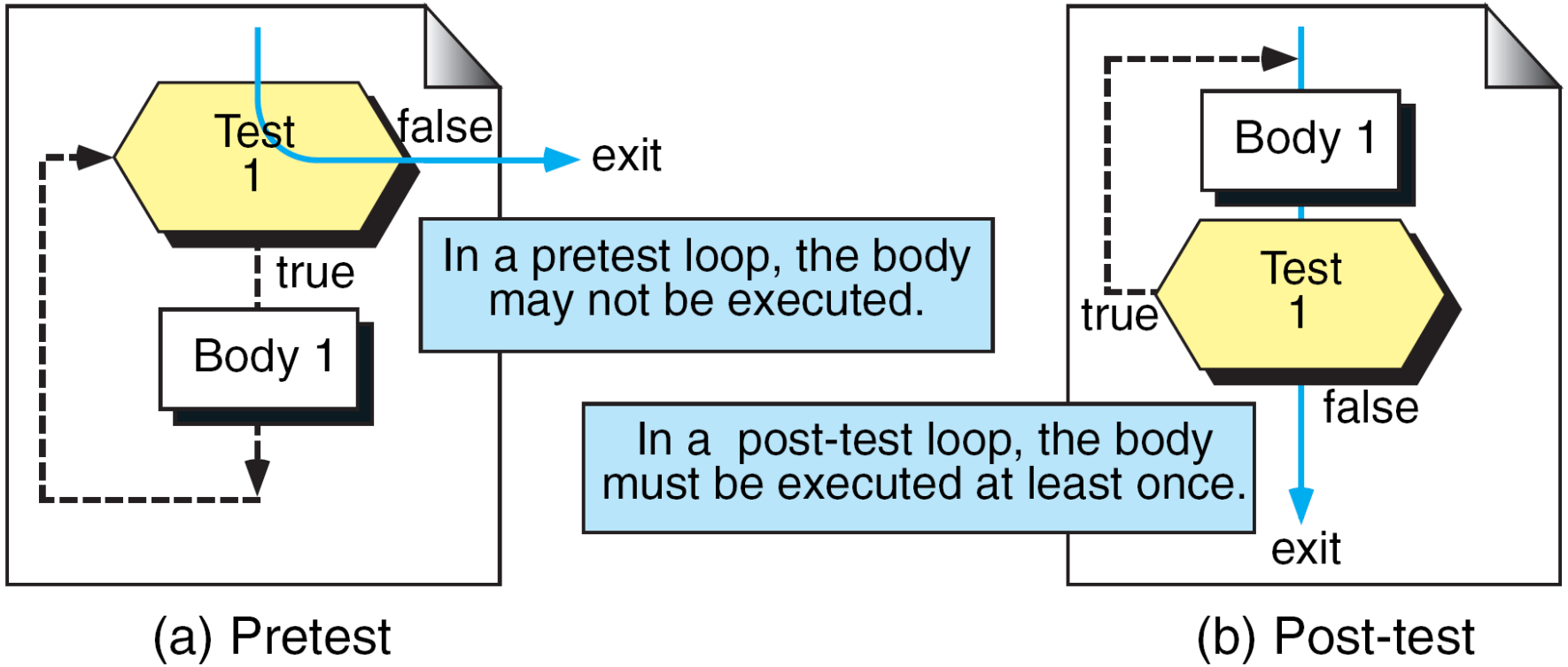


FIGURE 6-4 Minimum Number of Iterations in Two Loops

6-3 Initialization and Updating

In addition to the loop control expression, two other processes, initialization and updating, are associated with almost all loops.

Topics discussed in this section:

Loop Initialization
Loop Update

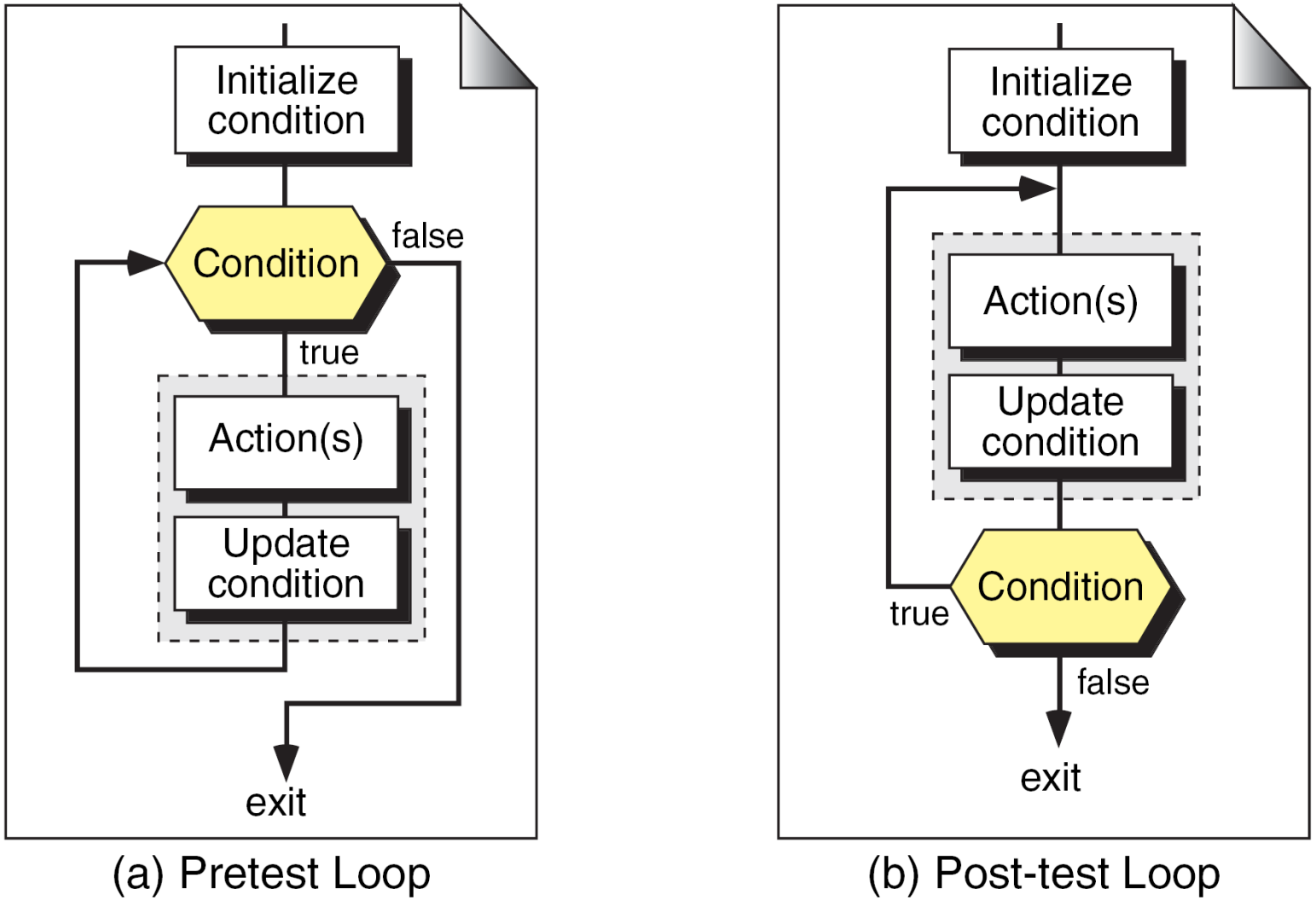


FIGURE 6-6 Initialization and Updating for Exercise

```c
#include <stdio.h>

int main () {

   /* Initialization */
   int a = 10;

   /* while loop execution */
   while( a < 20 ) {

      printf("value of a: %d\n", a);

      //Updating
      a = a +  1;
   }

   return 0;
}
```

How while loop works?

- The while loop evaluates the test expression inside the parenthesis ().
- If the test expression is true, statements inside the body of while loop are executed. Then, the test expression is evaluated again.
- The process goes on until the test expression is evaluated to false.
- If the test expression is false, the loop terminates (ends).

6-4 Event- and Counter-Controlled Loops

All the possible expressions that can be used in a loop limit test can be summarized into two general categories: event-controlled loops and counter-controlled loops.

Topics discussed in this section:

Event-Controlled Loops
Counter-Controlled Loops
Loop Comparison

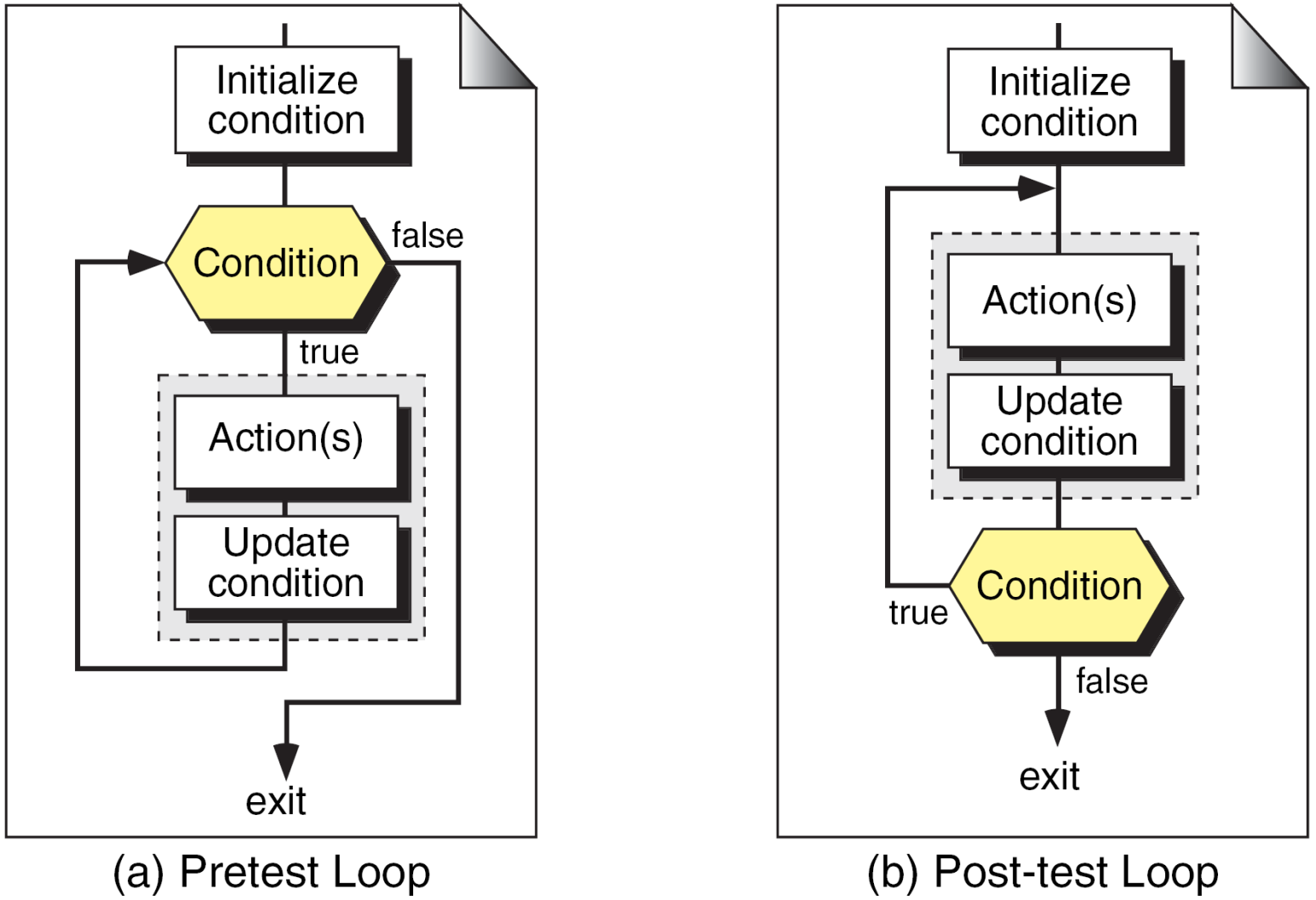


FIGURE 6-7 Event-controlled Loop Concept

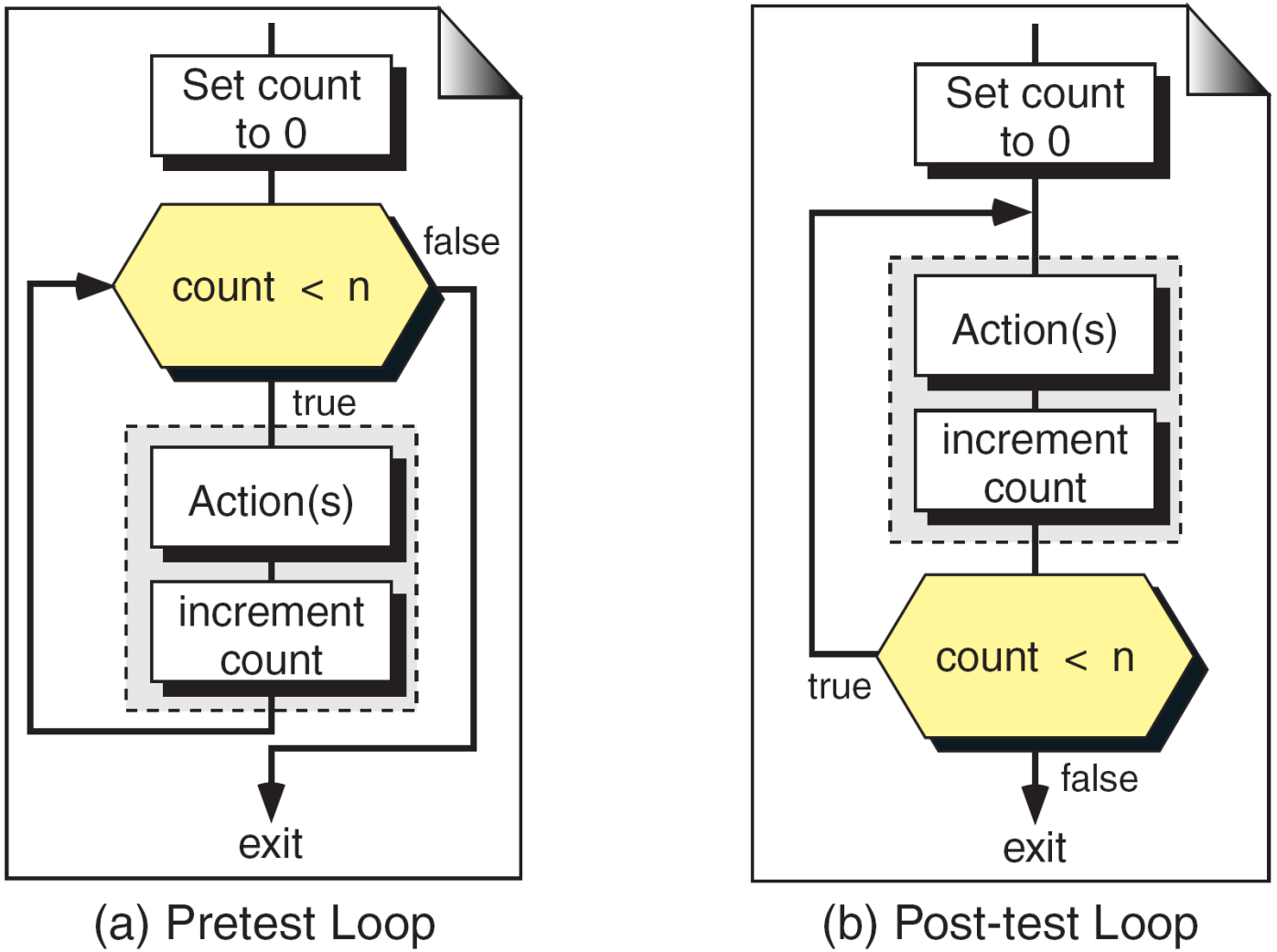


FIGURE 6-8 Counter-controlled Loop Concept

Pretest Loop		Post-test Loop	
Initialization:	1	Initialization:	1
Number of tests:	*n* + 1	Number of tests:	n
Action executed:	n	Action executed:	n
Updating executed:	n	Updating executed:	n
Minimum iterations:	0	Minimum iterations:	1

Table 6-1 Loop Comparisons

6-5 Loops in C

C has three loop statements: the while, the for, and the do…while. The first two are pretest loops, and the the third is a post-test loop. We can use all of them for event-controlled and counter-controlled loops.

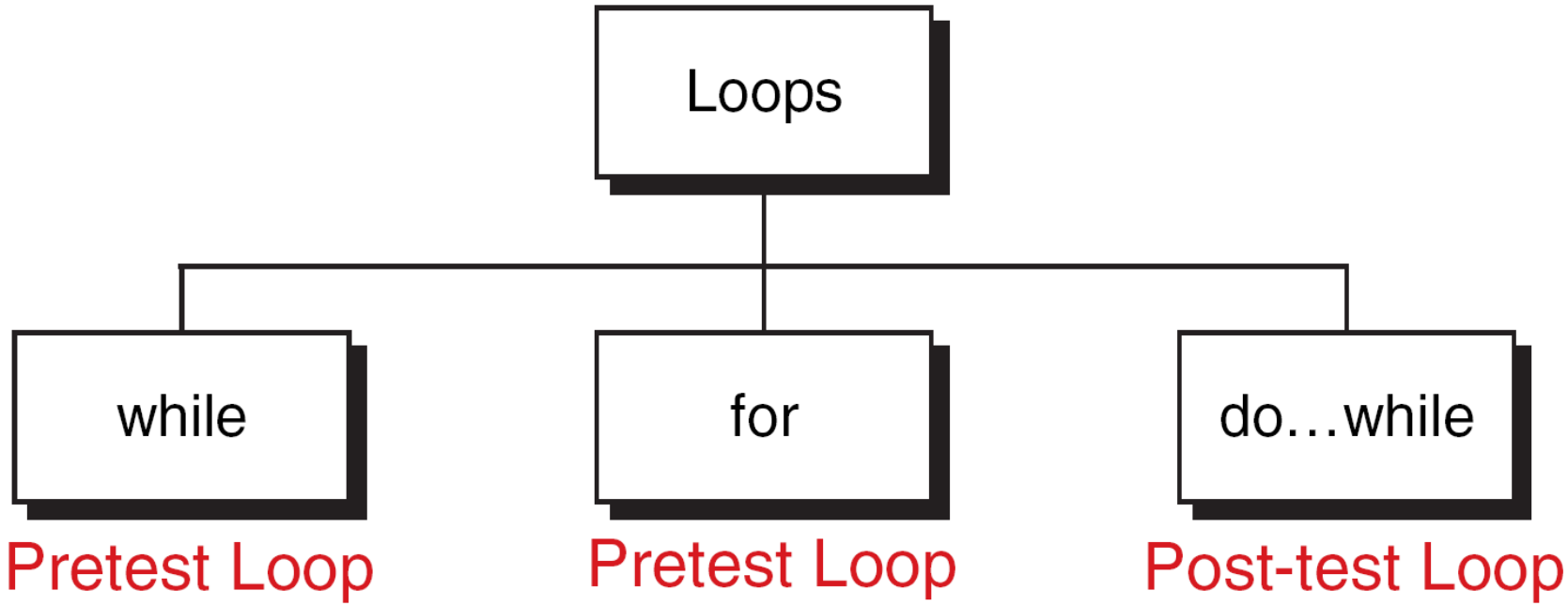
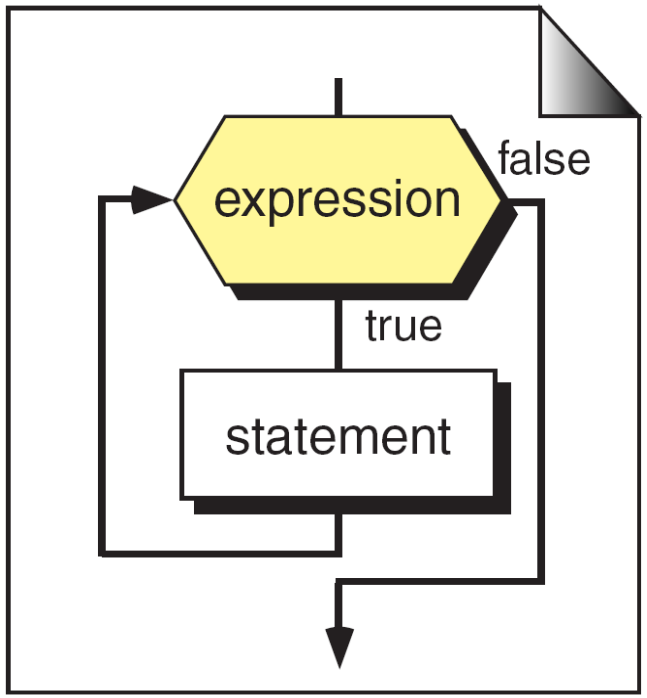


FIGURE 6-9 C Loop Constructs



(a) Flowchart

```
while (expression)

    statement
```

(b) Sample Code

FIGURE 6-10 The while Statement

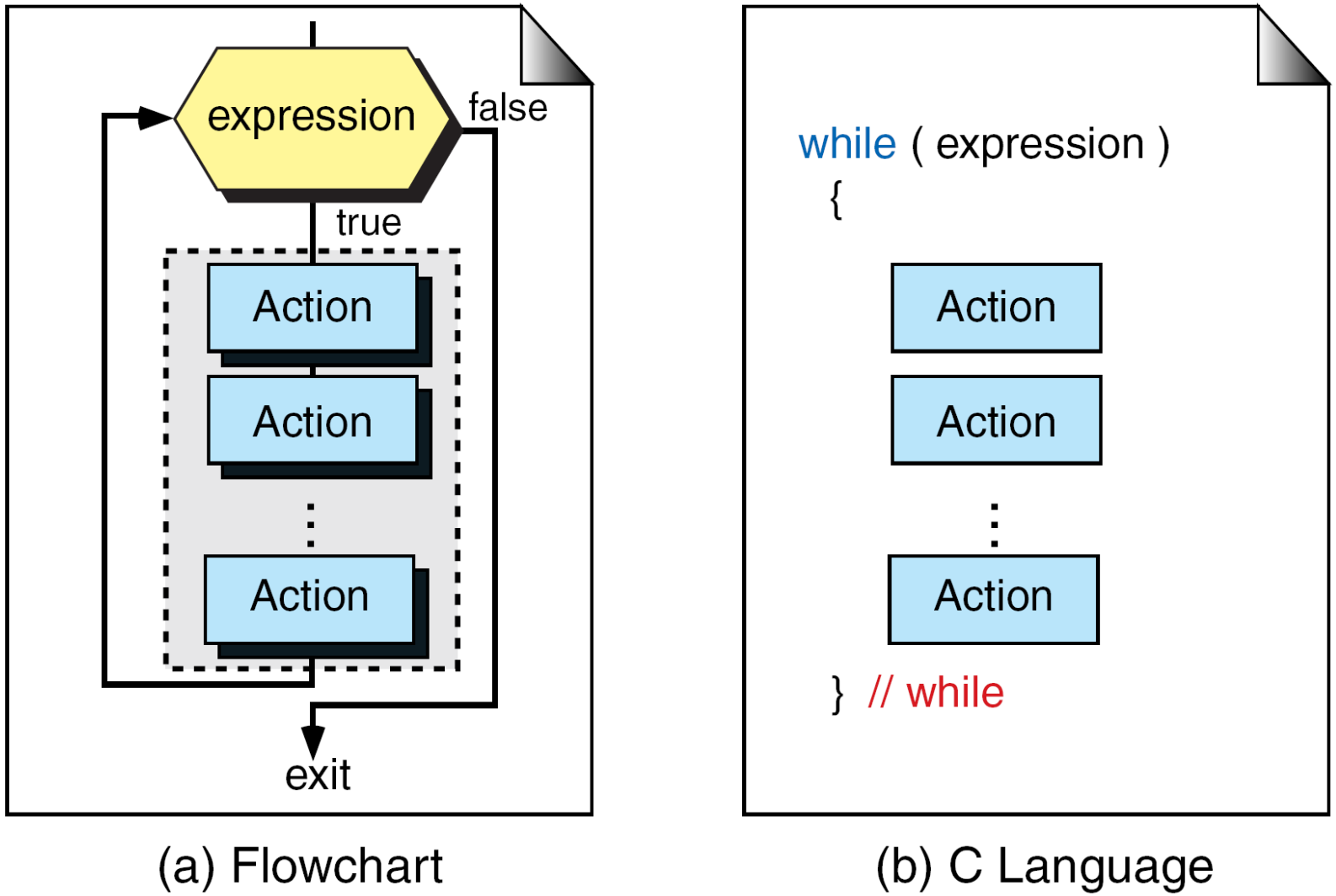


FIGURE 6-11 Compound while Statement