

Computer Science I

CS 135

6. Repetition: While and For Loops

a. Repetition Structures

b. While Loops

c. Do/While Loops

d. For Loops

- ✓ General form of a for loop
- ✓ Examples of for loops
- ✓ Nested for loops

6.d For Loops

General form of a for loop

➤ General flowchart of a for loop

- ✓ compared to a while loop, a for loop additionally contains an **initial statement** and an **update statement**

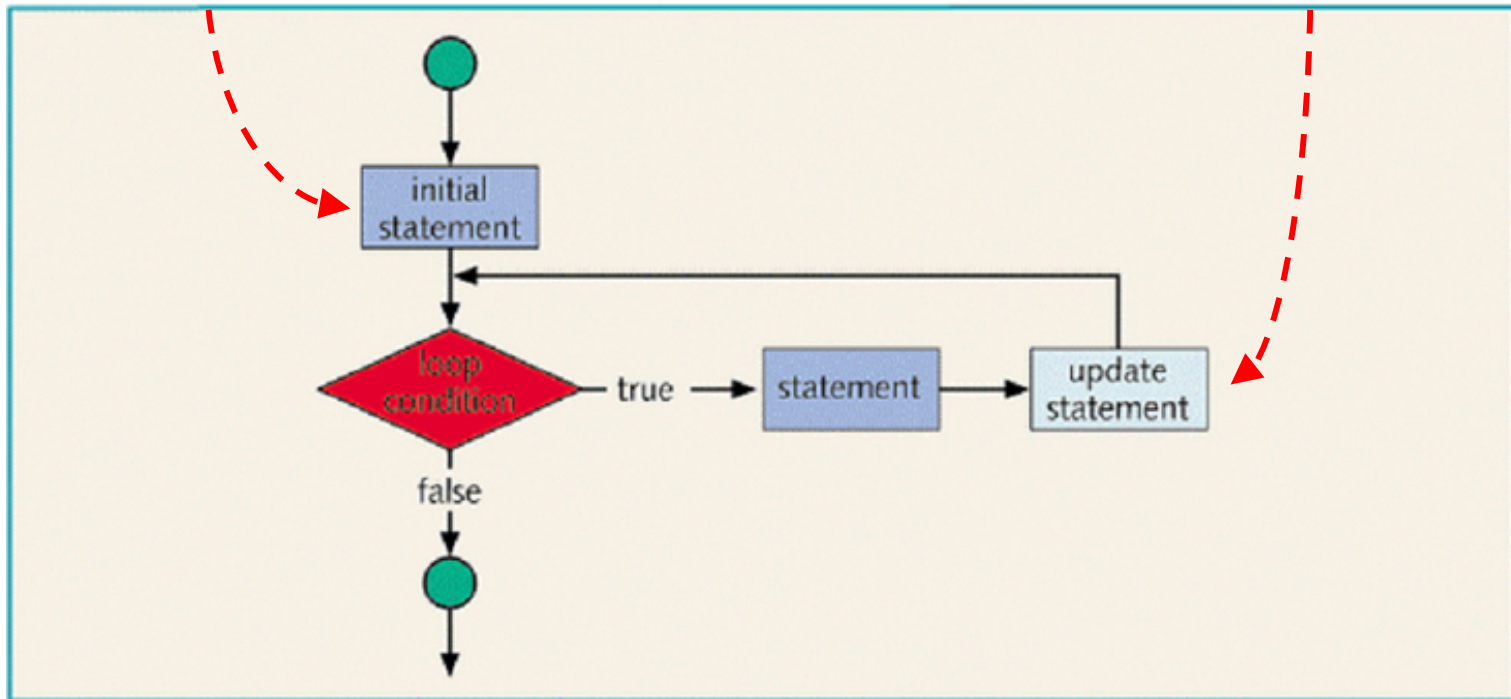


Figure 5-2 for loop

6.d For Loops

General form of a for loop

➤ General syntax of a for loop

```
for (i = 0; i < 10; i++) {  
    ...  
    cout << i;  
    ...  
}
```

```
for (initial statement; expression; update statement) {  
    statement1  
    statement2  
    statement3  
}
```

6.d For Loops

General form of a for loop

```
void main()
{
    // declare variables
    int num, sum = 0;
    int counter = 0;

    // prompt and increment sum 4 times
    while (counter < 4) {
        cout << "Enter number: ";
        cin >> num; sum += num;

        counter++;
    }

    // display result
    cout << "The sum is: ";
    cout << sum << endl;
}
```

```
void main()
{
    // declare variables
    int num, sum = 0;
    int counter = 0;

    // prompt and increment sum 4 times
    for (counter = 0; counter < 4;
         counter++) {
        cout << "Enter number: ";
        cin >> num; sum += num;

        counter++;
    }

    // display result
    cout << "The sum is: ";
    cout << sum << endl;
}
```

Implementing a counter-controlled loop with a for loop

6.d For Loops

General form of a for loop

- A for loop is really just a more compact way to write a while loop; it is *not* a new kind of control structure
 - ✓ it packages an initialization statement, a logical expression and an update statement on one line, the for header line
- A for loop executes as follows
 - ✓ the initial statement executes
 - ✓ the loop condition is evaluated
 - ✓ if the loop condition evaluates to true:
 - execute the loop body statements
 - execute the update statement
 - ✓ repeat previous step until the loop condition evaluates to false

6.d For Loops

General form of a for loop

➤ (Fun?) facts about for loops

- ✓ the initial statement generally initializes some variable
- ✓ the initial statement in the for loop is the first to be executed and is executed only once
- ✓ if the loop condition is initially false, the body statements never execute and the loop exits
- ✓ the update statement changes the value of the loop control variable which eventually sets the value of the condition to false
- ✓ the loop executes indefinitely if the loop condition stays true

6.d For Loops

General form of a for loop

➤ (Fun?) facts about for loops (cont'd)

- ✓ a semicolon at the end of the for line creates an empty loop: this is sometimes used as a (bad) way to slow down execution

```
for ( i = 0; i < 10000; i++ ) ;
```

- ✓ the *initial statement*, loop *condition*, and *update statement* may all be omitted, independently or together, for example:

```
for ( ; i < 10; i++ ) { ... }
```

```
for ( i = 0; ; i++ ) { ... }
```

```
for ( i = 0; i < 10; ) { ... }
```

```
for ( ; ; ) { ... }
```

- ✓ in this case, the flow of execution mostly depends on the body
→ *this is legal syntax but not necessarily good programming!*

6.d For Loops

Example of for loops

➤ For loops are mainly used as counter-controlled loops

- ✓ traditional counter-controlled loop

```
for (i = 0; i < max; i++) { ... }
```

- ✓ counter-controlled loop with different ranges

```
for (i = 1; i <= max; i++) { ... }
```

```
for (i = 7; i < max+7; i++) { ... }
```

- ✓ decreasing counter-controlled loops

```
for (i = max; i > 0; i--) { ... }
```

```
for (i = max-1; i >= 0; i--) { ... }
```


6.d For Loops

Example of for loops

➤ Other types of for loops are possible, but rarely used

- ✓ flag-controlled for loop

```
for (found = false; !found; found = ...) {  
    ...  
}
```

is the equivalent of

```
found = false;  
while (!found) {  
    ...  
    found = ...  
}
```

6.d For Loops

Nested for loops

- What code can print out a triangle of stars?

```
*  
* *  
* * *  
* * * *  
* * * * *
```

- ✓ answer:

```
for (i = 1; i <= 5 ; i++) {  
    for (j = 0; j < i; j++)  
        cout << "*" << " ";  
    cout << endl;  
}
```

6.d For Loops

Nested for loops

➤ What does this code print out?

```
for (i = 5; i >= 1 ; i--) {  
    for (j = 0; j < i; j++)  
        cout << "* ";  
    cout << endl;  
}
```

✓ answer:

```
* * * * *  
* * * *  
* * *  
* *  
*
```

6.d For Loops

Nested for loops

- What code can print out the identity matrix?

```
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1
```

```
✓ for (i = 0; i < 5 ; i++) {
    for (j = 0; j < i; j++) {
        cout << (i == j ? 1 : 0);
        cout << " ";
    }
    cout << endl;
}
```

Computer Science I

CS 135

6. Repetition: While and For Loops

a. Repetition Structures

b. While Loops

c. Do/While Loops

d. For Loops

- ✓ General form of a for loop
- ✓ Examples of for loops
- ✓ Nested for loops

Computer Science I

CS 135

6. Repetition: While and For Loops

- a. Repetition Structures
- b. While Loops
- c. Do/While Loops
- d. For Loops

Computer Science I

CS 135

0. Course Presentation
1. Introduction to Programming
2. Functions I: Passing by Value
3. File Input/Output
4. Predefined Functions
5. If and Switch Controls
6. While and For Loops
- 7. Functions II: Passing by Reference**
- 8. 1-D and 2-D Arrays**