

JavaScript: Control Statements II

Outline

1. **Introduction**
2. **Essentials of Counter-Controlled Repetition**
3. **for Repetition Statement**
4. **Examples Using the for Statement**
5. **switch Multiple-Selection Statement**
6. **do...while Repetition Statement**
7. **break and continue Statements**
8. **Labeled break and continue Statements**
9. **Logical Operators**
10. **Summary of Structured Programming**

Objectives

- In this lesson, you will learn:
 - To be able to use the **for** and **do...while** repetition statements to execute statements in a program repeatedly.
 - To understand multiple selection using the **switch** selection statement.
 - To be able to use the **break** and **continue** program-control statements.
 - To be able to use the logical operators.

1 Introduction

- Continuation of Chapter 8
 - Theory and principles of structured programming

2 Essentials of Counter-Controlled Repetition

- Counter-controlled repetition
 - Name of a control
 - Initial value
 - Increment or decrement
 - Final value

Outline

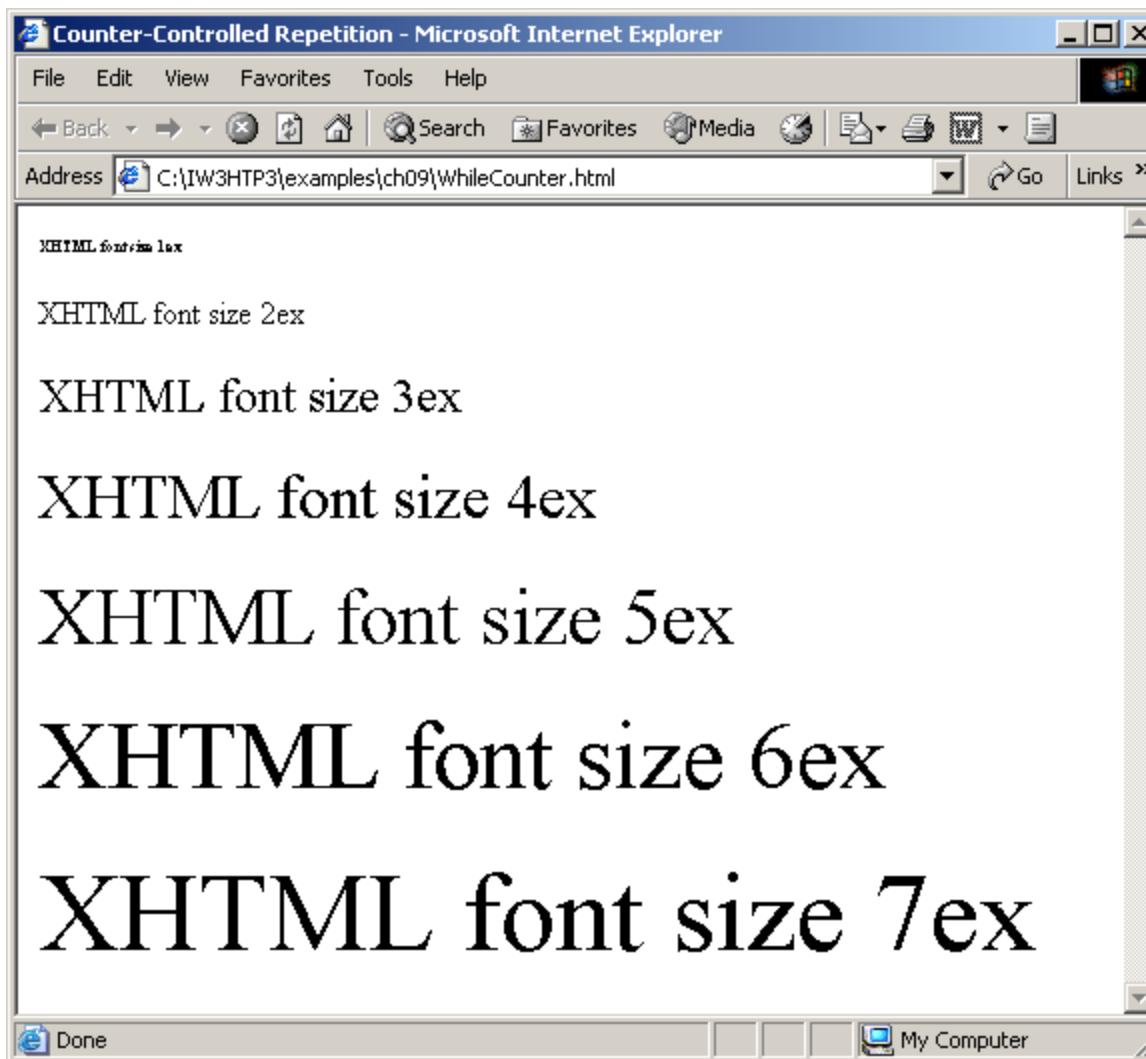
WhileCounter.html (1 of 2)

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 9.1: WhileCounter.html -->
6 <!-- Counter-Controlled Repetition -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Counter-Controlled Repetition</title>
11
12   <script type = "text/javascript">
13     <!--
14       var counter = 1;           // initialization
15
16       while ( counter <= 7 ) {    // repetition condition
17         document.writeln( "<p style = \"font-size: " +
18           counter + "ex\">XHTML font size " + counter +
19           "ex</p>" );
20         ++counter;               // increment
21     }
22     // -->
23   </script>
24
```

25 </head><body>

26 </html>

Outline



**WhileCounter.html
(2 of 2)**

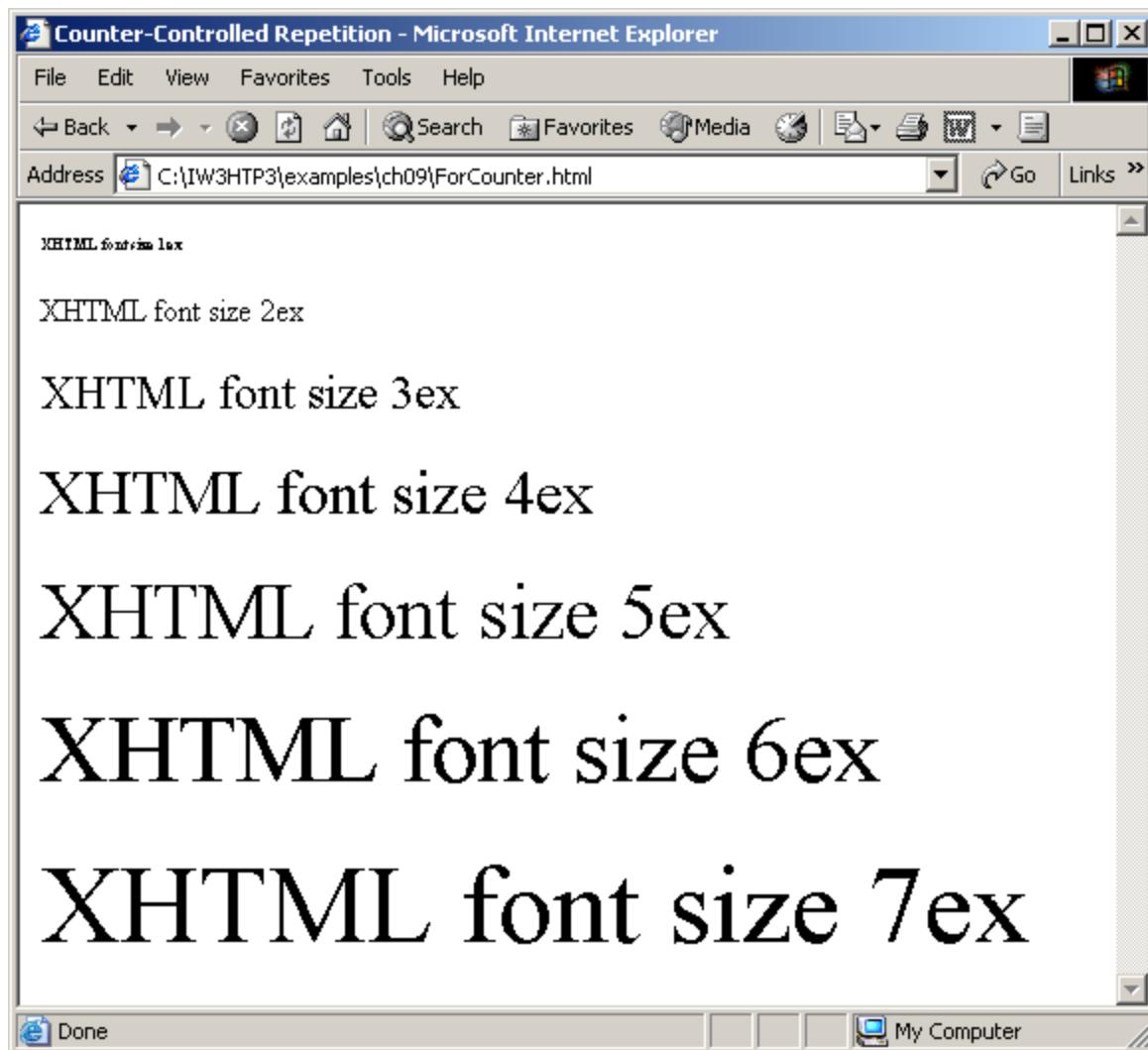
3 for Repetition Statement

- **for** repetition statement
 - Handles all the details of counter-controlled repetition
 - **for** structure header
 - The first line

Outline

ForCounter.html (1 of 1)

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 9.2: ForCounter.html                               -->
6 <!-- Counter-Controlled Repetition with for statement -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Counter-Controlled Repetition</title>
11
12   <script type = "text/javascript">
13     <!--
14       // Initialization, repetition condition and
15       // incrementing are all included in the for
16       // statement header.
17       for ( var counter = 1; counter <= 7; ++counter )
18         document.writeln( "<p style = \"font-size: " +
19                           counter + "ex\">"XHTML font size " + counter +
20                           "ex</p>" );
21     // -->
22   </script>
23
24   </head><body></body>
25 </html>
```

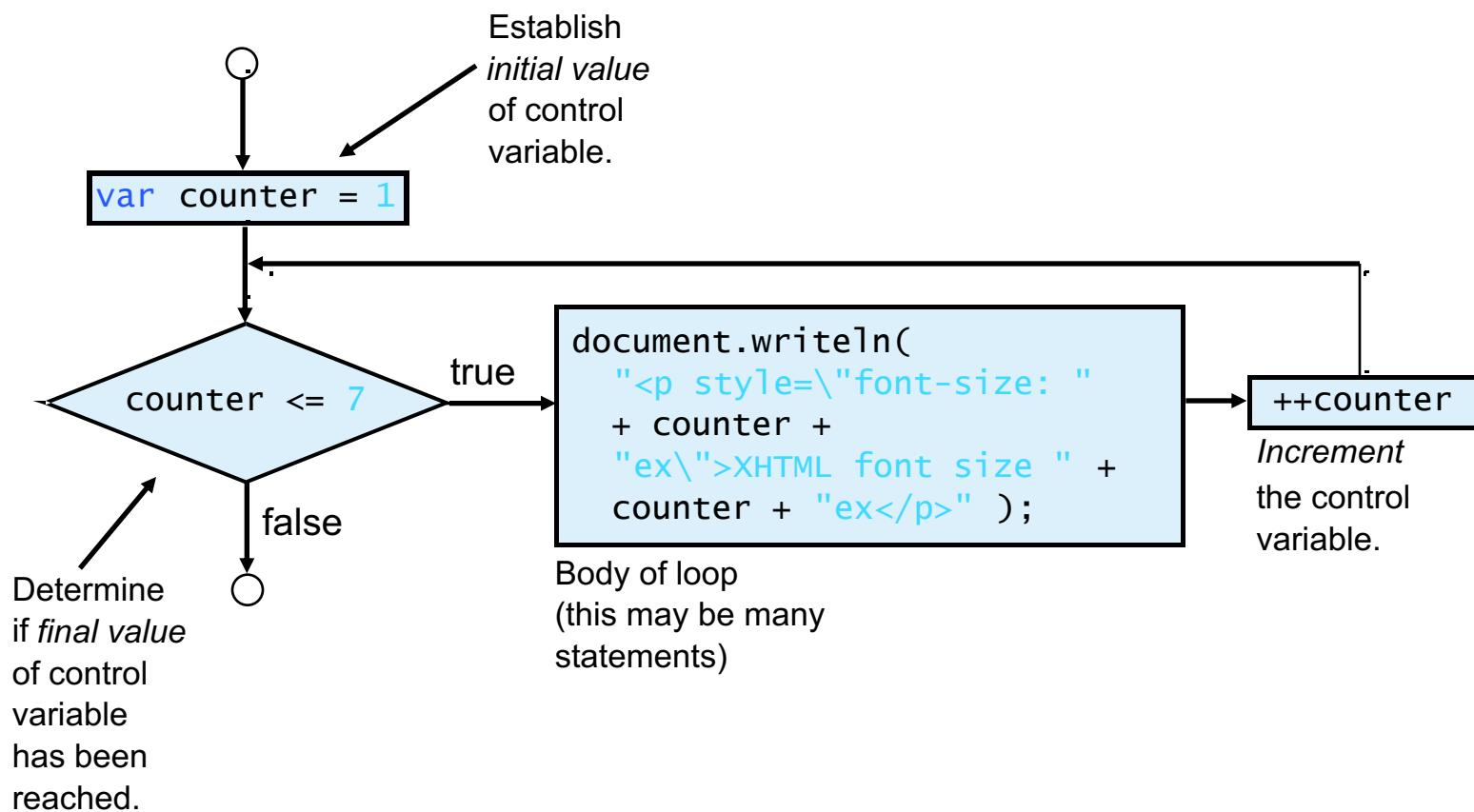


3 for Repetition Statement

```
for keyword           Control variable name           Final value of control variable  
for ( var counter = 1; counter <= 7; ++counter )  
Initial value of control variable   Loop-continuation condition   for which the condition is true  
Increment of control variable
```

The diagram illustrates the structure of a for loop. It shows the keyword 'for' followed by a block of code in parentheses. The code consists of three parts separated by semicolons: an initialization statement, a loop-continuation condition, and an increment statement. Arrows point from descriptive text labels to their corresponding parts in the code. The 'for keyword' points to the first 'for'. The 'Control variable name' points to 'counter'. The 'Final value of control variable for which the condition is true' points to '7'. The 'Initial value of control variable' points to '1'. The 'Loop-continuation condition' points to the condition 'counter <= 7'. The 'Increment of control variable' points to '+counter'.

3 for Repetition Statement



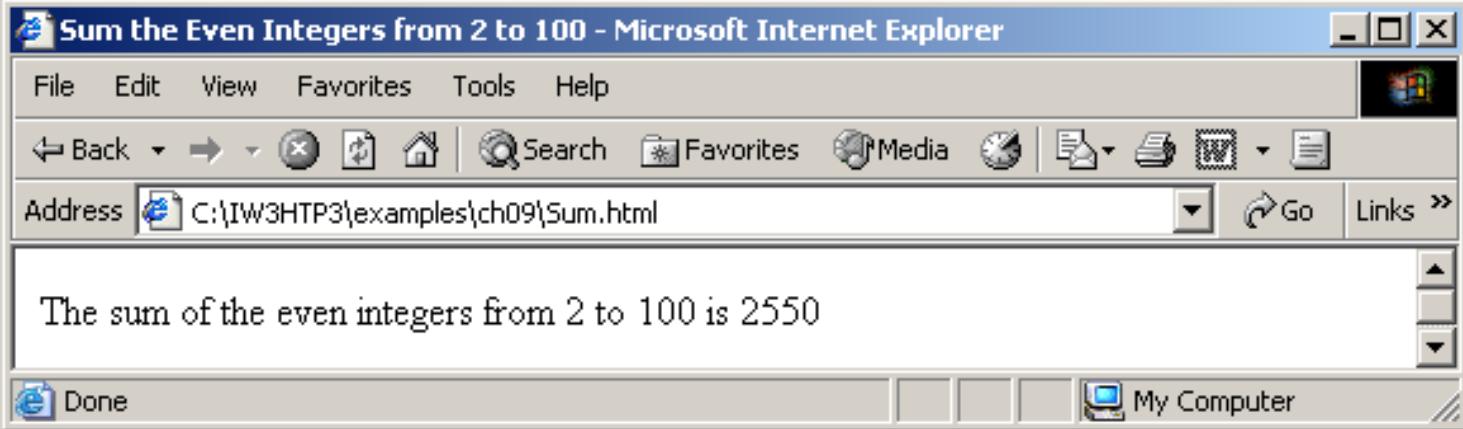
4 Examples Using the `for` Statement

- Summation with `for`
- Compound interest calculation with `for` loop
 - `Math` object
 - Method `pow`
 - Method `round`

Outline

Sum.html (1 of 1)

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 9.5: Sum.html -->
6 <!-- Using the for repetition statement -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Sum the Even Integers from 2 to 100</title>
11
12   <script type = "text/javascript">
13     <!--
14       var sum = 0;
15
16       for ( var number = 2; number <= 100; number += 2 )
17         sum += number;
18
19       document.writeln( "The sum of the even integers " +
20         "from 2 to 100 is " + sum );
21       // -->
22     </script>
23
24   </head><body></body>
25 </html>
```



Outline

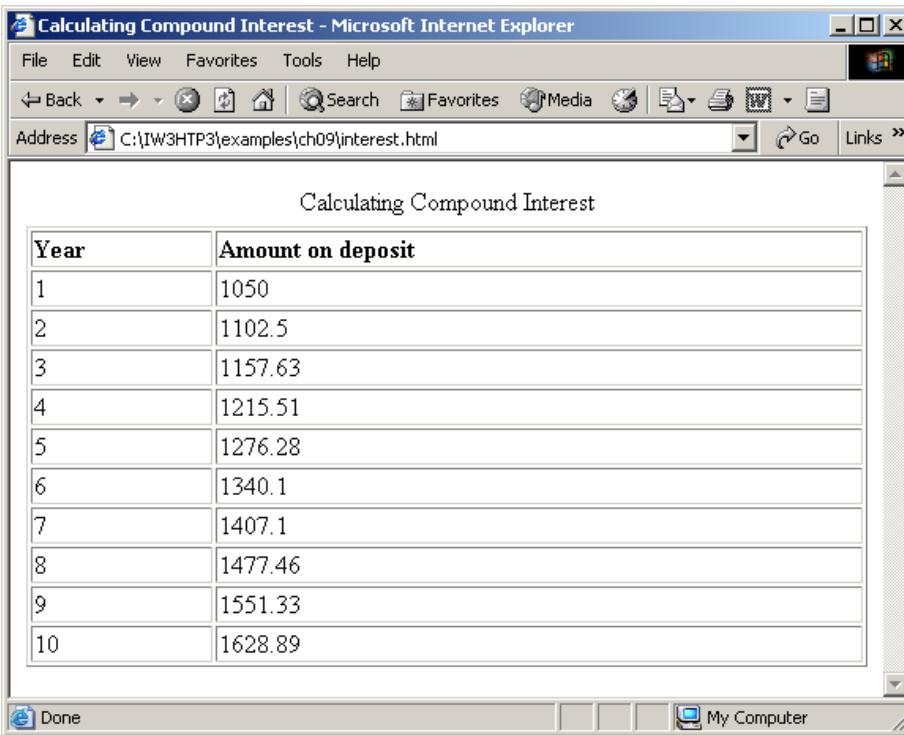
Interest.html (1 of 2)

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 9.6: Interest.html          -->
6 <!-- Using the for repetition statement -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Calculating Compound Interest</title>
11
12   <script type = "text/javascript">
13     <!--
14       var amount, principal = 1000.0, rate = .05;
15
16       document.writeln(
17         "<table border = \"1\" width = \"100%\">" );
18       document.writeln(
19         "<caption>Calculating Compound Interest</caption>" );
20       document.writeln(
21         "<thead><tr><th align = \"left\">"Year</th>" );
22       document.writeln(
23         "<th align = \"left\">"Amount on deposit</th>" );
24       document.writeln( "</tr></thead>" );
25
```

Outline

Interest.html (2 of 2)

```
26 for ( var year = 1; year <= 10; ++year ) {
27     amount = principal * Math.pow( 1.0 + rate, year );
28     document.writeln( "<tbody><tr><td>" + year +
29                         "</td><td>" + Math.round( amount * 100 ) / 100 +
30                         "</td></tr>" );
31 }
32
33 document.writeln( "</tbody></table>" );
34 // -->
35 </script>
36
37 </head><body></body>
38 </html>
```



A screenshot of Microsoft Internet Explorer version 6.0 displaying a table titled "Calculating Compound Interest". The table shows the growth of a deposit over 10 years at a 5% annual interest rate, compounded annually. The browser interface includes a menu bar, toolbar, address bar, and status bar.

Year	Amount on deposit
1	1050
2	1102.5
3	1157.63
4	1215.51
5	1276.28
6	1340.1
7	1407.1
8	1477.46
9	1551.33
10	1628.89

5 switch Multiple-Selection Statement

- Controlling expression
- Case labels
- Default case

Outline

SwitchTest.html (1 of 3)

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 9.7: SwitchTest.html -->
6 <!-- Using the switch statement -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Switching between XHTML List Formats</title>
11
12   <script type = "text/javascript">
13     <!--
14       var choice,           // user's choice
15           startTag,        // starting list item tag
16           endTag,          // ending list item tag
17           validInput = true, // indicates if input is valid
18           listType;         // list type as a string
19
20       choice = window.prompt( "Select a list style:\n" +
21         "1 (bullet), 2 (numbered), 3 (lettered)", "1" );
22
```

Outline

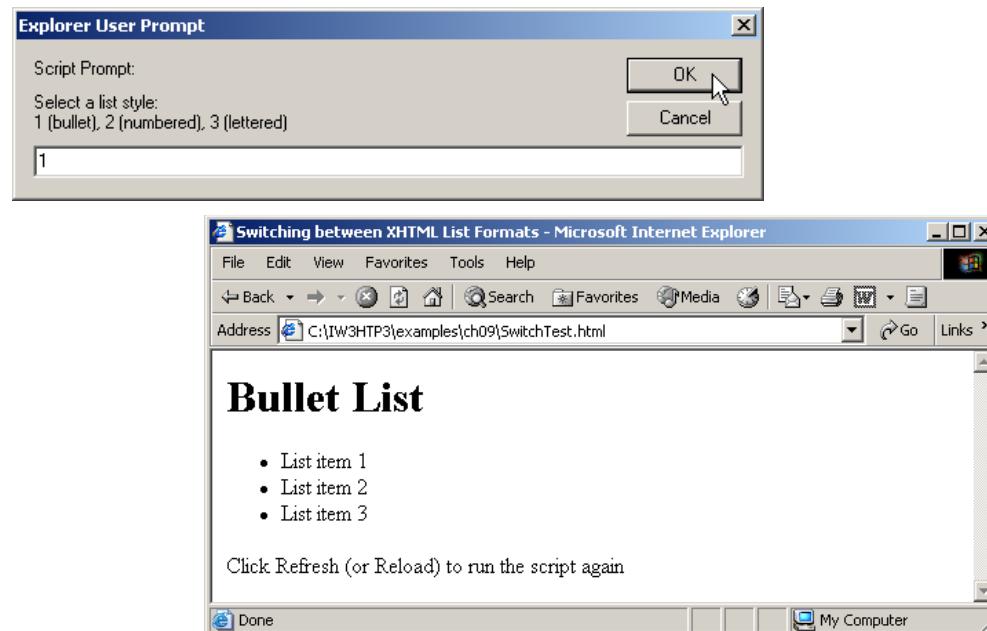
SwitchTest.html (2 of 3)

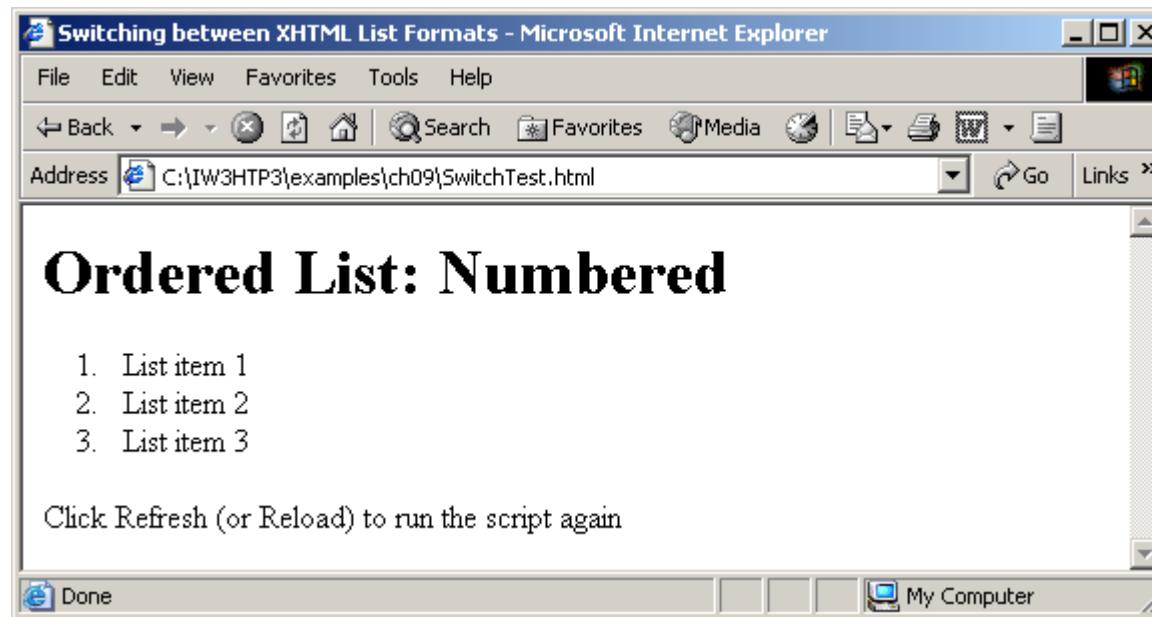
```
23 switch ( choice ) {
24
25     case "1":
26         startTag = "<ul>";
27         endTag = "</ul>";
28         listType = "<h1>Bullet List</h1>";
29         break;
30
31     case "2":
32         startTag = "<ol>";
33         endTag = "</ol>";
34         listType = "<h1>Ordered List: Numbered</h1>";
35         break;
36
37     case "3":
38         startTag = "<ol type = \"A\">";
39         endTag = "</ol>";
40         listType = "<h1>Ordered List: Lettered</h1>";
41         break;
42
43     default:
44         validInput = false;
45
46     }
47
48
49     if ( validInput == true ) {
50
51         document.writeln( listType + startTag );
52
53
54         for ( var i = 1; i <= 3; ++i )
55             document.writeln( "<li>List item " + i + "</li>" );
56
57 }
```

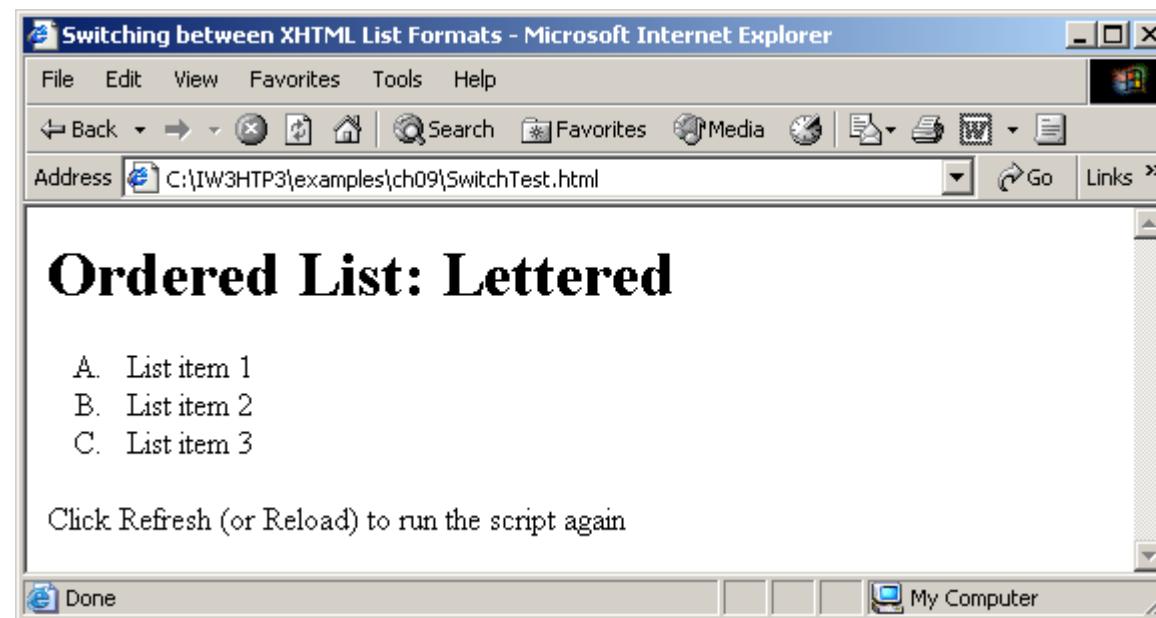
Outline

SwitchTest.html (3 of 3)

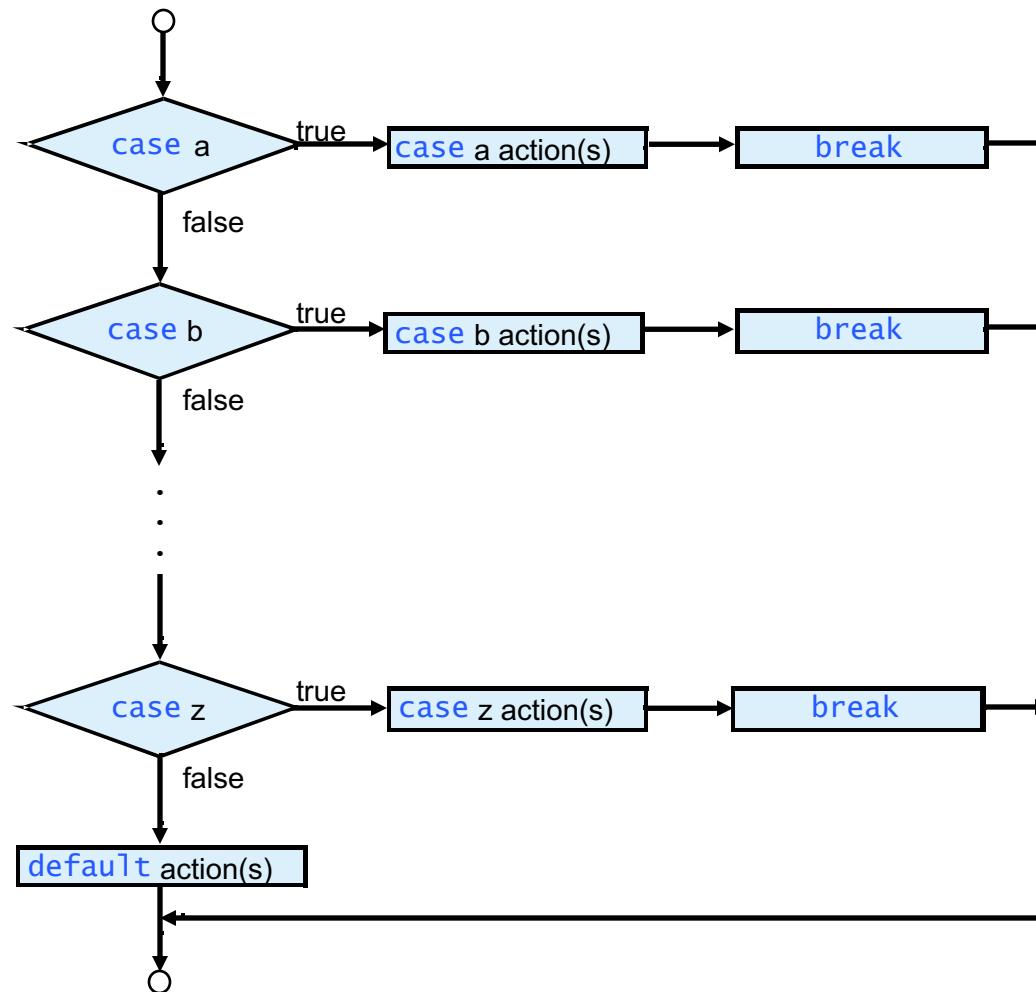
```
48
49     document.writeln( endTag );
50 }
51 else
52     document.writeln( "Invalid choice: " + choice );
53 // -->
54 </script>
55
56 </head>
57 <body>
58     <p>Click Refresh (or Reload) to run the script again</p>
59 </body>
60 </html>
```







5 switch Multiple-Selection Statement



6 do...while Repetition Statement

- Similar to the `while` statement
- Tests the loop continuation condition after the loop body executes
- Loop body always executes at least once

Outline

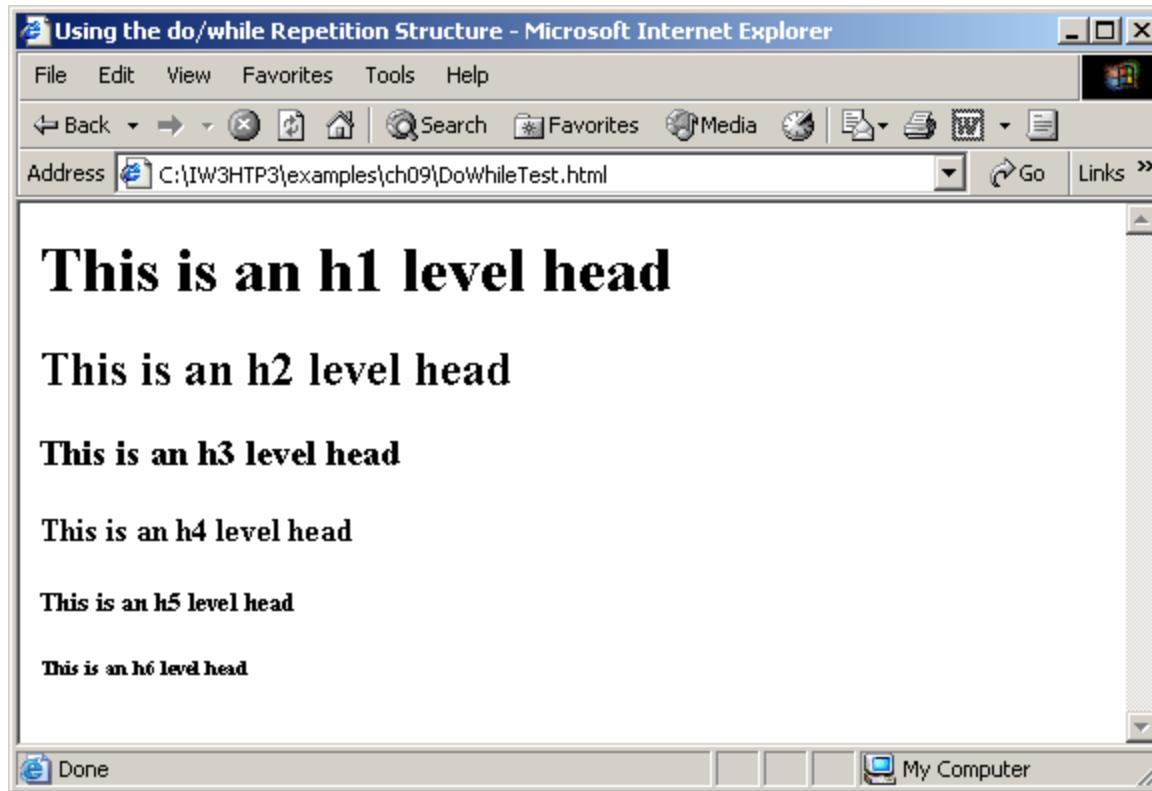
DoWhileTest.html (1 of 2)

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 9.9: DoWhileTest.html      -->
6 <!-- Using the do...while statement -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Using the do...while Repetition Statement</title>
11
12   <script type = "text/javascript">
13     <!--
14       var counter = 1;
15
16       do {
17         document.writeln( "<h" + counter + ">This is " +
18           "an h" + counter + " level head" + "</h" +
19           counter + ">" );
20
21         ++counter;
22     } while ( counter <= 6 );
23 // -->
24   </script>
```

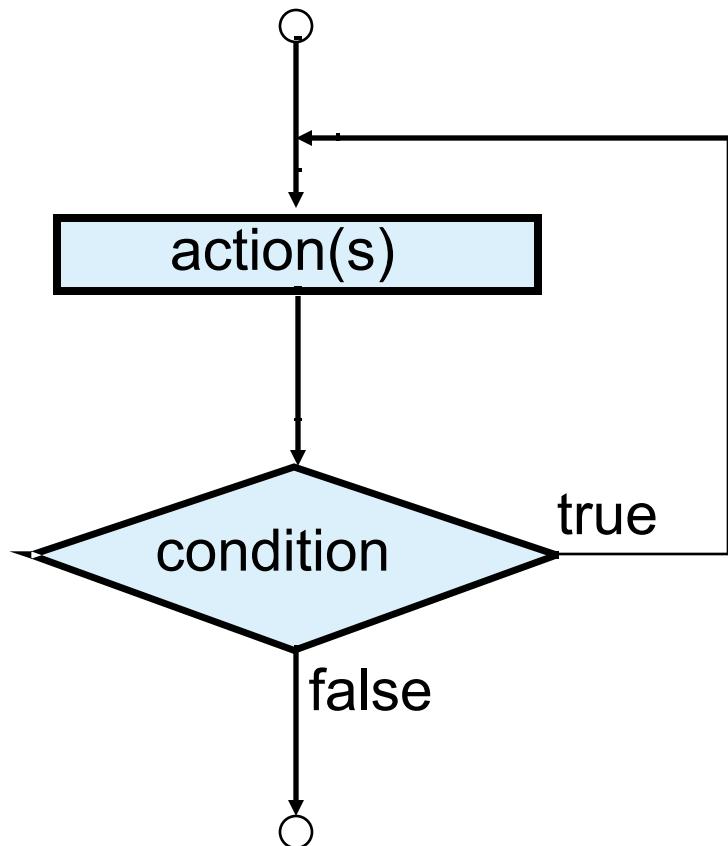
```
25  
26    </head><body></body>  
27  </html>
```

Outline

DoWhileTest.html (2 of 2)



6 do...while Repetition Structure



7 break and continue Statements

- **break**
 - Immediate exit from the structure
 - Used to escape early from a loop
 - Skip the remainder of a `switch` statement
- **continue**
 - Skips the remaining statements in the body of the structure
 - Proceeds with the next iteration of the loop

Outline

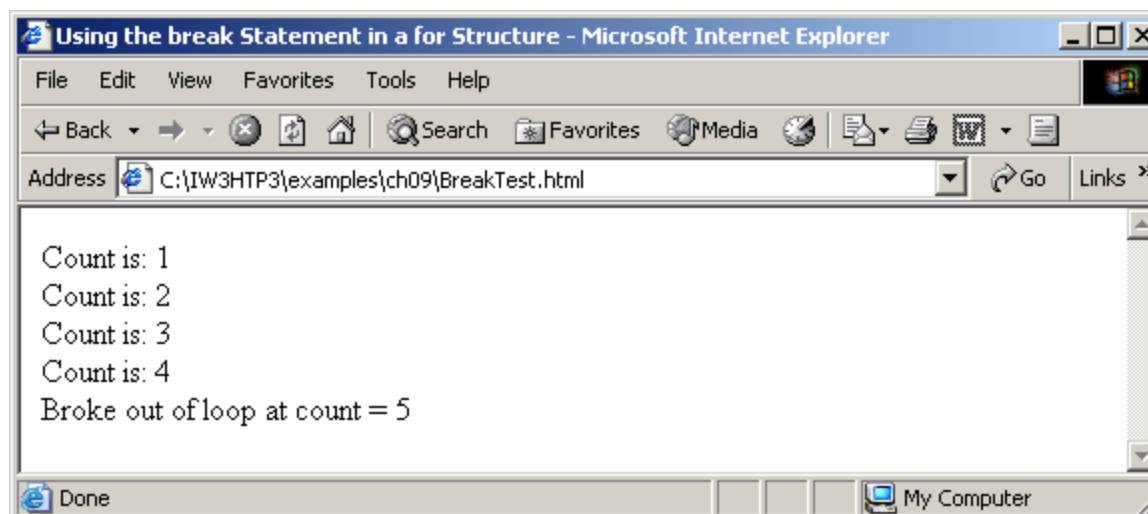
BreakTest.html (1 of 2)

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 9.11: BreakTest.html -->
6 <!-- Using the break statement -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>
11      Using the break Statement in a for Structure
12    </title>
13
14   <script type = "text/javascript">
15     <!--
16       for ( var count = 1; count <= 10; ++count ) {
17         if ( count == 5 )
18           break; // break loop only if count == 5
19
20         document.writeln( "Count is: " + count + "<br />" );
21     }
22   </script>
```

Outline

BreakTest.html (2 of 2)

```
23 document.writeln(  
24     "Broke out of loop at count = " + count );  
25 // -->  
26 </script>  
27  
28 </head><body></body>  
29 </html>
```



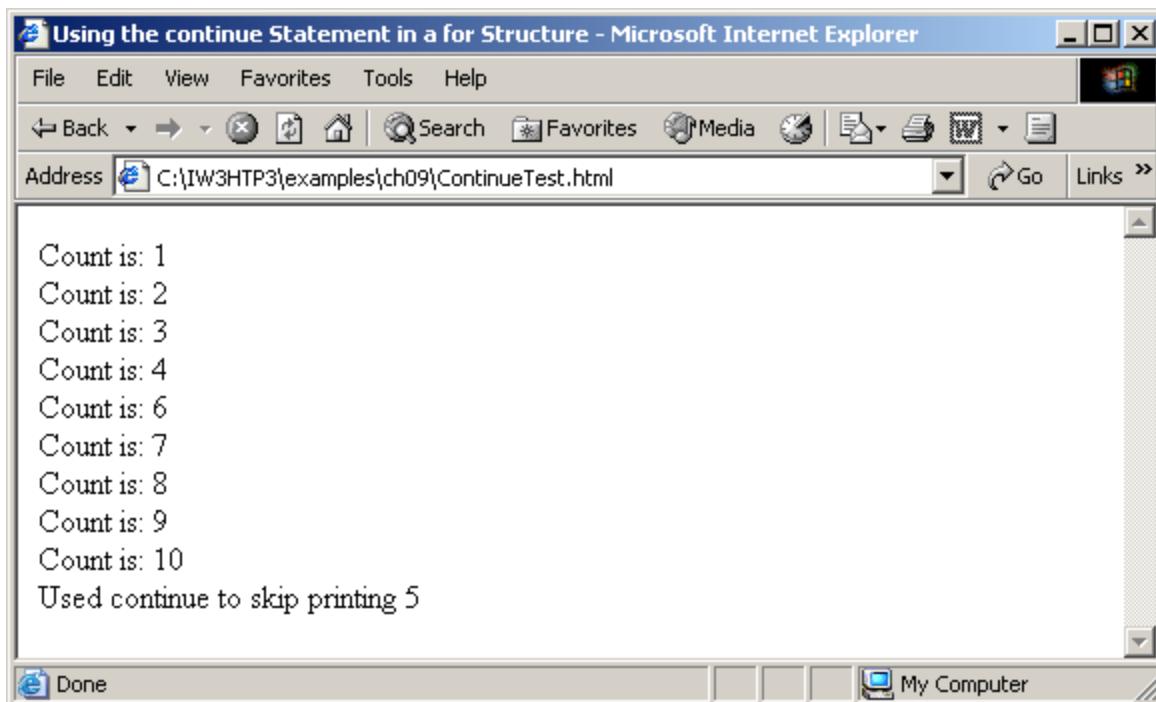
Outline

ContinueTest.html (1 of 2)

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 9.12: ContinueTest.html -->
6 <!-- Using the break statement -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>
11      Using the continue Statement in a for Structure
12    </title>
13
14   <script type = "text/javascript">
15     <!--
16       for ( var count = 1; count <= 10; ++count ) {
17         if ( count == 5 )
18           continue; // skip remaining code in loop
19           // only if count == 5
20
21         document.writeln( "Count is: " + count + "<br />" );
22     }
23
```

Outline

ContinueTest.html (2 of 2)



8 Labeled break and continue Statements

- **Labeled break statement**
 - Break out of a nested set of structures
 - Immediate exit from that structure and enclosing repetition structures
 - Execution resumes with first statement after enclosing labeled statement
- **Labeled continue statement**
 - Skips the remaining statements in structure's body and enclosing repetition structures
 - Proceeds with next iteration of enclosing labeled repetition structure
 - Loop-continuation test evaluates immediately after the **continue** statement executes

Outline

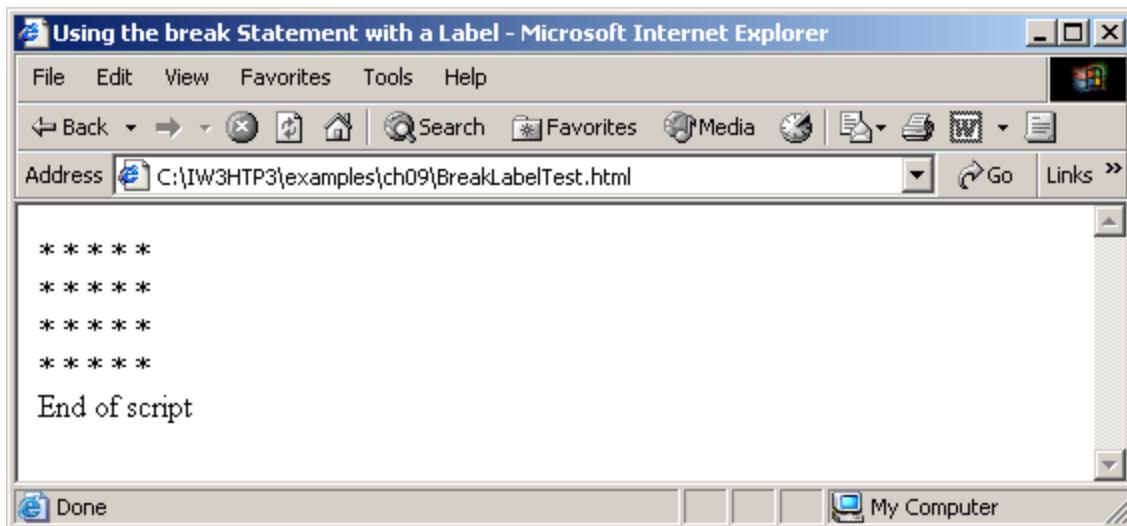
BreakLabelTest.html (1 of 2)

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 9.13: BreakLabelTest.html          -->
6 <!-- Using the break statement with a Label -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Using the break Statement with a Label</title>
11
12   <script type = "text/javascript">
13     <!--
14       stop: { // labeled block
15         for ( var row = 1; row <= 10; ++row ) {
16           for ( var column = 1; column <= 5 ; ++column ) {
17
18             if ( row == 5 )
19               break stop; // jump to end of stop block
20
21             document.write( " * " );
22           }
23
24           document.writeln( "<br />" );
25         }
```

Outline

BreakLabelTest.html (2 of 2)

```
26
27     // the following line is skipped
28     document.writeln( "This line should not print" );
29 }
30
31     document.writeln( "End of script" );
32     // -->
33 </script>
34
35 </head><body></body>
36 </html>
```



Outline

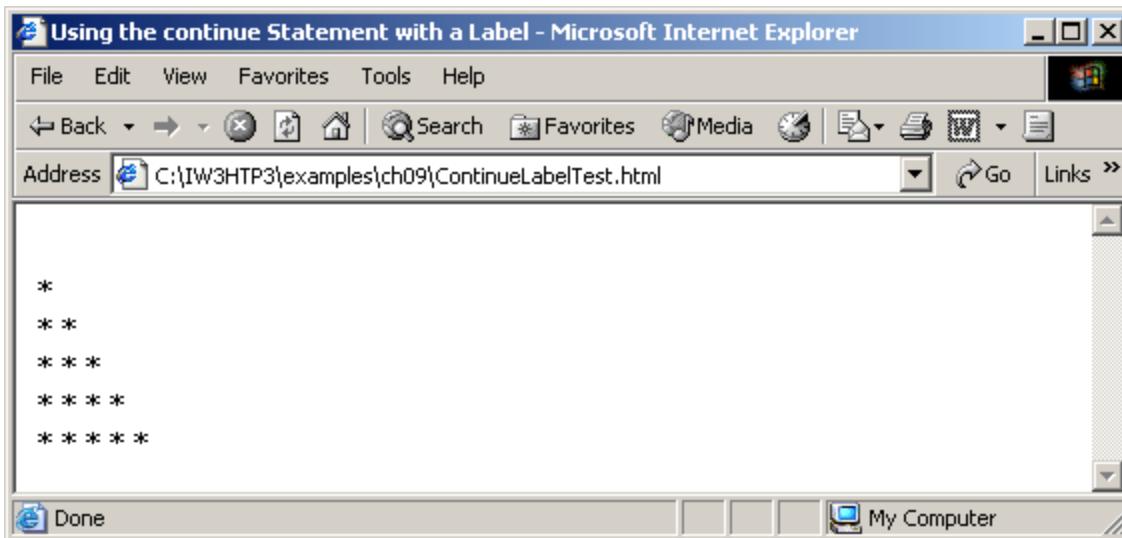
ContinueLabelTest.html (1 of 2)

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 9.14: ContinueLabelTest.html -->
6 <!-- Using the continue statement      -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Using the continue Statement with a Label</title>
11
12   <script type = "text/javascript">
13     <!--
14       nextRow: // target label of continue statement
15       for ( var row = 1; row <= 5; ++row ) {
16         document.writeln( "<br />" );
17
18         for ( var column = 1; column <= 10; ++column ) {
19
20           if ( column > row )
21             continue nextRow; // next iteration of
22                         // labeled loop
23
24           document.write( "* " );
25     }
```

```
26      }
27      // -->
28  </script>
29
30  </head><body></body>
31 </html>
```

Outline

ContinueLabelTest.html (2 of 2)



9 Logical Operators

- More logical operators
 - Logical AND (`&&`)
 - Logical OR (`||`)
 - Logical NOT (`!`)

9 Logical Operators

expression1	expression2	expression1 && expression2
false	false	false
false	true	false
true	false	false
true	true	true

9 Logical Operators

expression1	expression2	expression1 expression2
false	false	false
false	true	true
true	false	true
true	true	true

expression	!expression
false	true
true	false

Outline

LogicalOperators.html (1 of 2)

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 9.18: LogicalOperators.html -->
6 <!-- Demonstrating Logical Operators -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Demonstrating the Logical Operators</title>
11
12   <script type = "text/javascript">
13     <!--
14       document.writeln(
15         "<table border = \"1\" width = \"100%\">" );
16
17       document.writeln(
18         "<caption>Demonstrating Logical " +
19         "Operators</caption" );
20
21       document.writeln(
22         "<tr><td width = \"25%\">Logical AND (&&)</td>" +
23         "<td>false && false: " + ( false && false ) +
24         "<br />false && true: " + ( false && true ) +
25         "<br />true && false: " + ( true && false ) +
```

Outline

LogicalOperators.html
(2 of 2)

```
26    "<br />true && true: " + ( true && true ) +
27    "</td>" );
28
29    document.writeln(
30        "<tr><td width = \"25%\">Logical OR (||)</td>" +
31        "<td>false || false: " + ( false || false ) +
32        "<br />false || true: " + ( false || true ) +
33        "<br />true || false: " + ( true || false ) +
34        "<br />true || true: " + ( true || true ) +
35        "</td>" );
36
37    document.writeln(
38        "<tr><td width = \"25%\">Logical NOT (!)</td>" +
39        "<td>!false: " + ( !false ) +
40        "<br />!true: " + ( !true ) + "</td>" );
41
42    document.writeln( "</table>" );
43    // -->
44    </script>
45
46    </head><body></body>
47 </html>
```

Demonstrating the Logical Operators - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media

Address C:\IW3HTP3\examples\ch09\LogicalOperators.html Go Links

Demonstrating Logical Operators

Logical AND (&&)	false && false: false false && true: false true && false: false true && true: true
Logical OR ()	false false: false false true: true true false: true true true: true
Logical NOT (!)	!false: true !true: false

Done My Computer

9 Logical Operators

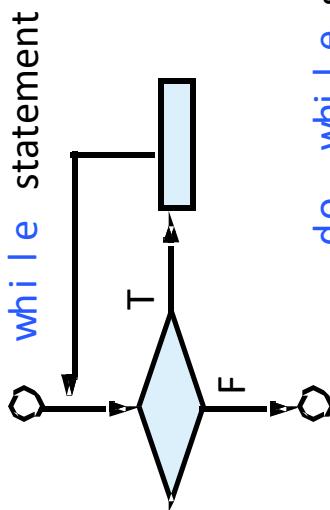
Operator	Associativity	Type
<code>++ -- !</code>	right to left	unary
<code>* / %</code>	left to right	multiplicative
<code>+ -</code>	left to right	additive
<code>< <= > >=</code>	left to right	relational
<code>== !=</code>	left to right	equality
<code>&&</code>	left to right	logical AND
<code> </code>	left to right	logical OR
<code>? :</code>	right to left	conditional
<code>= += -= *= /= %=</code>	right to left	assignment

10 Summary of Structured Programming

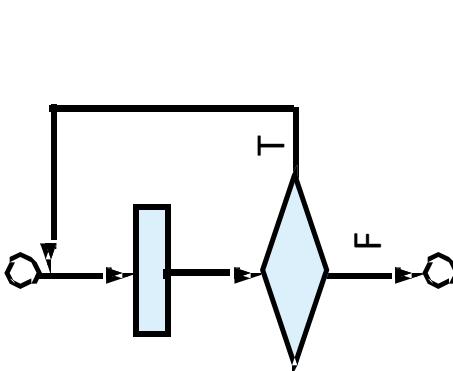
- Flowcharts
 - Reveal the structured nature of programs
- Single-entry/single-exit control structures
 - Only one way to enter and one way to exit each control structure
- Control structure stacking
 - The exit point of one control structure is connected to the entry point of the next control structure

10 Summary of Structured Programming

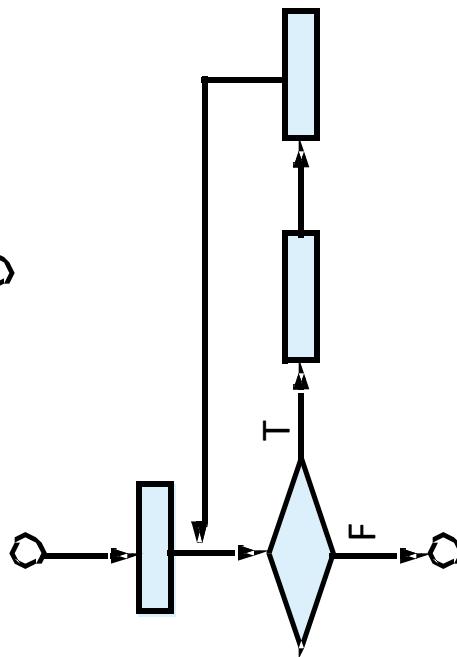
Repetition



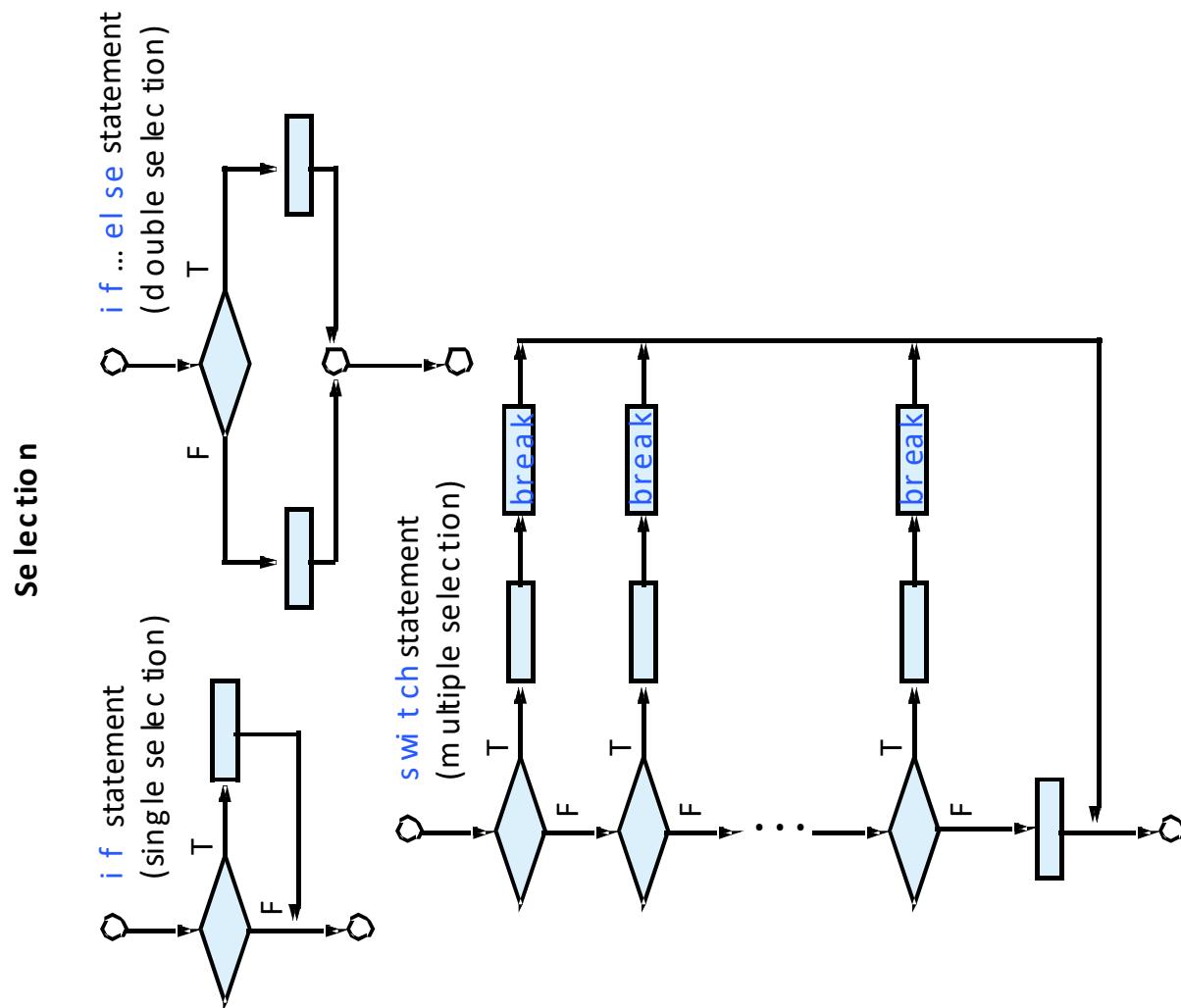
do...while statement



for statement

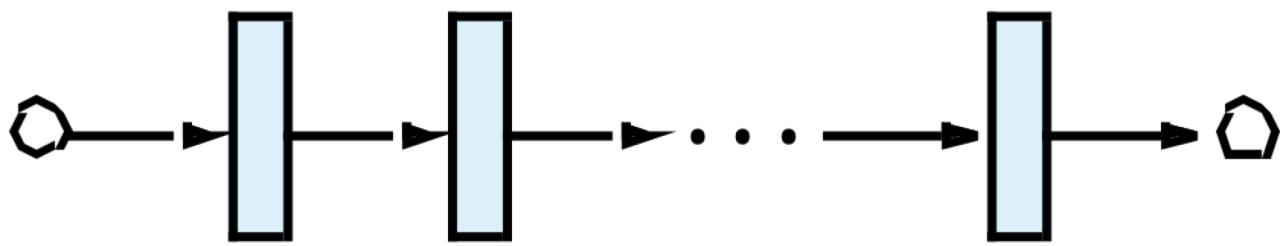


10 Summary of Structured Programming



10 Summary of Structured Programming

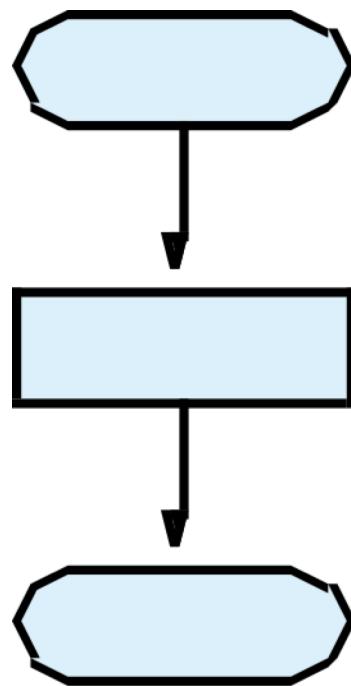
Se que nce



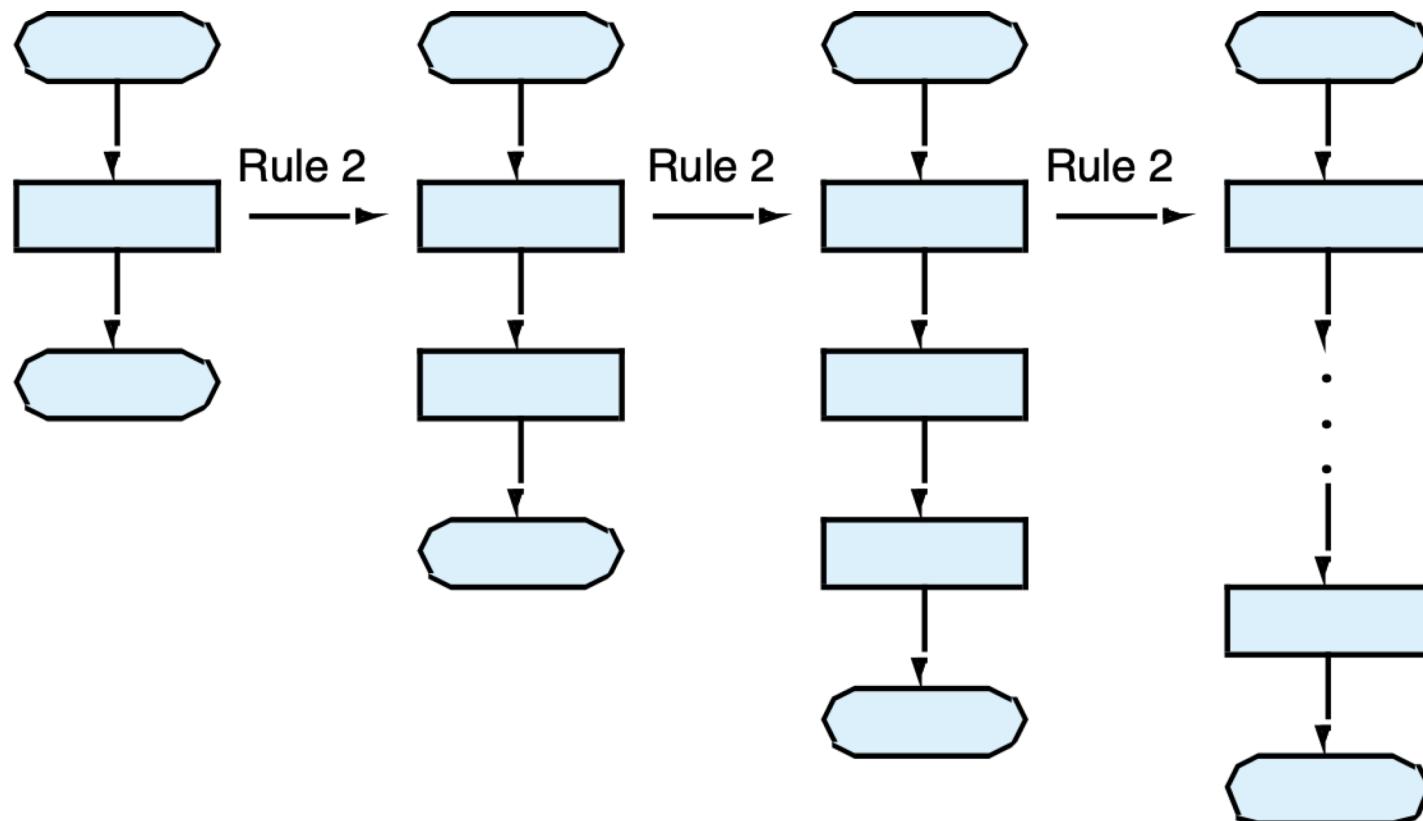
10 Summary of Structured Programming

Rules for Forming Structured Programs	
1)	Begin with the “simplest flowchart” (Fig. 9.22).
2)	Any rectangle (action) can be replaced by two rectangles (actions) in sequence.
3)	Any rectangle (action) can be replaced by any control structure (sequence, <code>if</code> , <code>if...else</code> , <code>switch</code> , <code>while</code> , <code>do...while</code> or <code>for</code>).
4)	Rules 2 and 3 may be applied as often as you like and in any order.

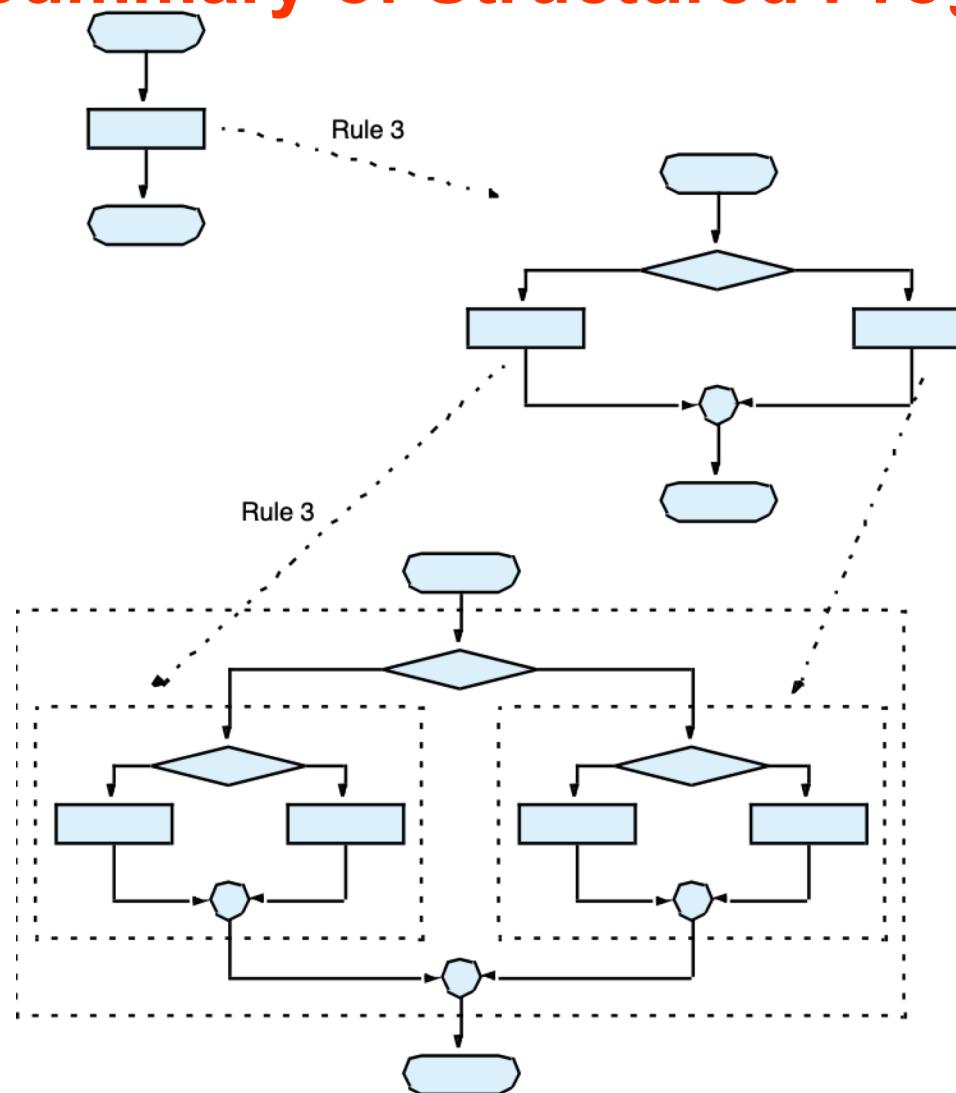
10 Summary of Structured Programming



10 Summary of Structured Programming

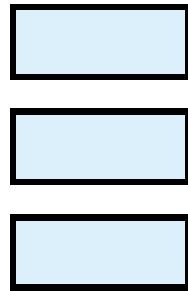


10 Summary of Structured Programming

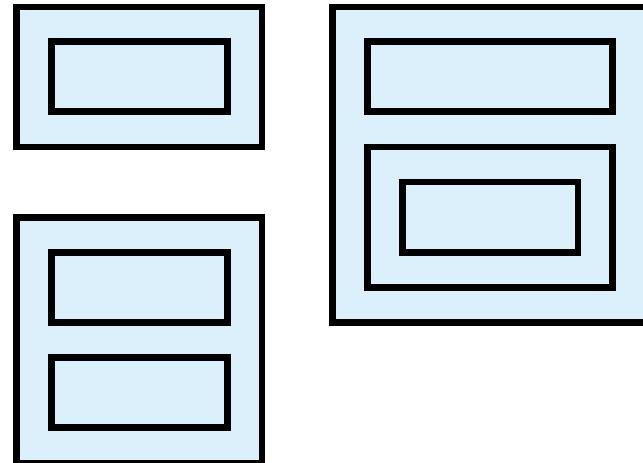


10 Summary of Structured Programming

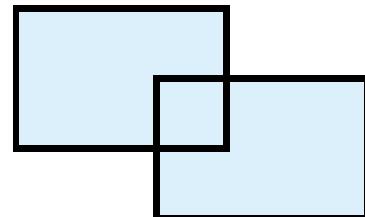
Stacked building blocks



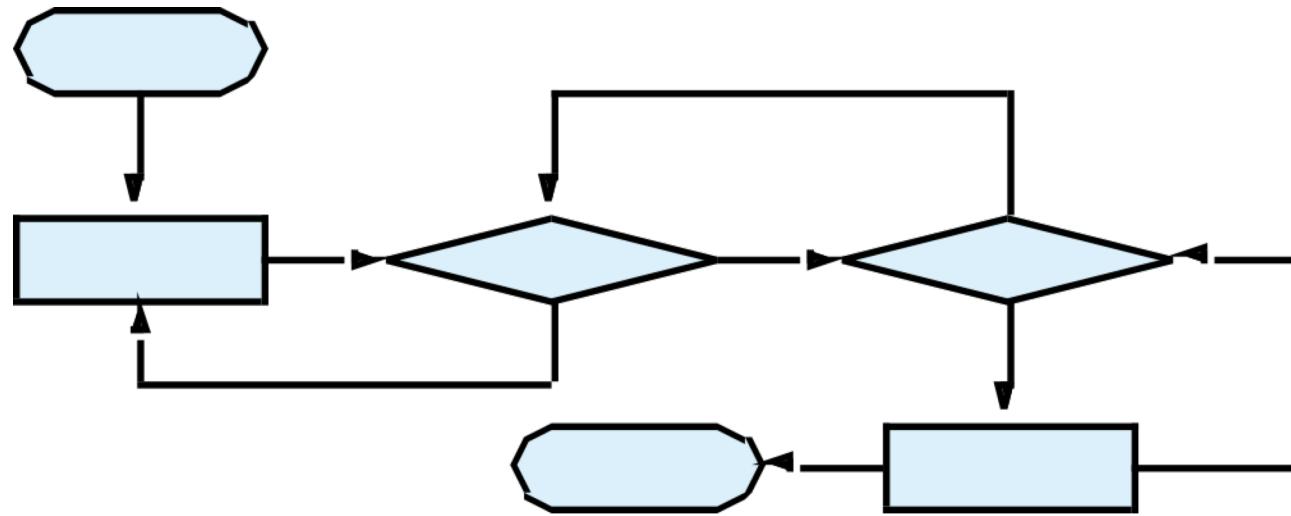
Nested building blocks



Overlapping building blocks
(Illegal in structured programs)



10 Summary of Structured Programming



Web Resources

- www.javascriptmall.com
- developer.netscape.com/tech/javascript
- www.mozilla.org/js/language
- Deitel and deitel, Internet and World Wide Web How to Program:
Third Edition