More Operators and Control Structures

- Logical Operators (true/false)
  - AND &&
  - OR ||
  - NOT !
- break and continue keywords
- More Control Structures
  - do - while repetition structure
  - for repetition strucure
  - switch - case selection structure

 1

Logical Operators

- Used in while and if assertions true/false
- There are three logical operators
  - AND &&
  - OR ||
  - NOT !

A	B	A && B
F	F	F
F	T	F
T	F	F
T	T	T

A	B	A || B
F	F	F
F	T	T
T	F	T
T	T	T

A	!A
F	T
T	F

Note on Precedence: Evaluate relational first and then logical

 2

Operators Precedence

(Highest to Lowest)

.	Property access of an object
( )	Defines order of operation
- ++ --	Minus, Increment, Decrement
!	Logical NOT Operator
* / %	Multiply, Division, Remainder
+ -	Addition, Subtraction
< <= > >=	} Relational Operators
== !=	
&&	Logical AND Operator
||	Logical OR Operator
= += -= *= /= %=	Compound Assignment

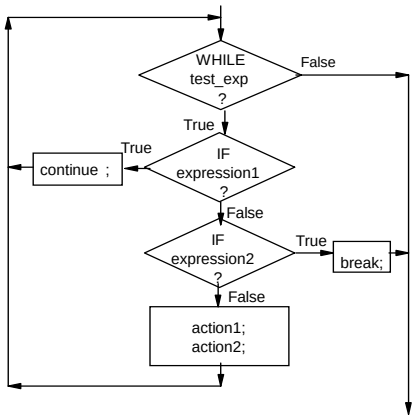
 3

Logical Operator Examples

```
if(A==B && A==C)
if(!Valid)
if(A = 0)     // Error use ==
if(!(A || B))
if(!A && !B)
A <= B || C == D
A = B == 0;
if(Question == "C" || Question == "c")
if(SSN > 999999999 || SSN < 0)
if(Tax == 0 || Tax ==  15 || Tax == 28)
```

 4

break; continue; *commands*




```
while(test_exp)
{
    if(expression1)
      continue;
    if(expression2)
      break;
    action1;
    action2;
}
```

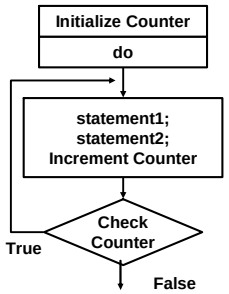
 5

Filtered Input Application 2

```
<head>
  <title>Filtered Data Entry</title>
  <script type="text/javascript">
    var Entry;
    while(true)
    {
       Entry = window.prompt( "Do you like Programming? (y or n)", "" );
       if(Entry == "y" || Entry == "Y")  {
         document.writeln("<h2>I\'m glad you like programming!</h2>");
         break;
       }
       else if(Entry == "n" || Entry == "N")  {
         document.writeln("<h2>You will like it if you study.</h2>");
         break;
       }
       else
         window.alert("You must enter either y or n !");
    }
  </script>
 </head>
 <body>   <p>Click Refresh (or Reload) to run the script again</p> </body>
```

do - while Structure

- A loop structure that guarantees the loop body is executed once.
- Condition is tested at bottom of loop
- Don't forget the semicolon for while(...);



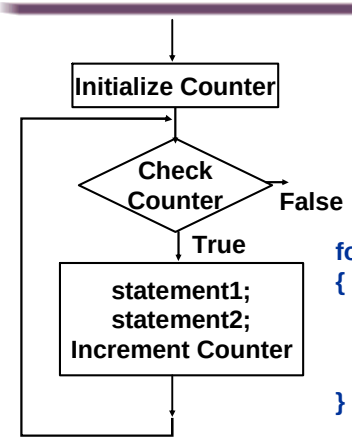

```
Initialize Counter;
do
{
        statement1;
        statement2;
        Increment Counter;
}while(Check Counter);
```

 7

Filtered Input Application 3

```
<head>
  <title>Filtered Data Entry</title>
  <script type="text/javascript">
    var Entry;
    do
    {
      Entry = window.prompt( "Do you like Programming?", "y or n" );
    }while(!(Entry=="y" || Entry=="Y" || Entry=="n" || Entry=="N"));
    if( Entry == "y" || Entry == "Y" )
      document.writeln("<h2>I\'m glad you like programming!</h2>");
    else
      document.writeln("<h2>You will like it if you study.</h2>");
  </script>
 </head>
 <body>
   <p>Click Refresh (or Reload) to run the script again</p>
 </body>
```

for loop Flow Chart




```
for(Initialize;Check;Increment)
{
        statement1;
        statement2;
        statement3;
}
```

 9

for Loop Example

```
<head>
  <title>Average Calculation 2</title>
  <script type = "text/javascript">
   var Score = 0, ScoreTotal = 0, Count = 0;
   for(Count = 1; Count <= 5; Count++)
   {
     Score = parseInt(window.prompt("Enter Score",""));
     ScoreTotal = ScoreTotal + Score;
     document.writeln("Score " + Count + " = " + Score + "<br/>");
   }
   document.writeln("<h2>The Average Score = "
         + ScoreTotal / 5 + "</h2>");
  </script>
 </head>
 <body>
  <p>Click Refresh (or Reload) to run the script again</p>
 </body>
```

 10

XHTML: Tables

- Tables display information in tabular form
  - `<table> </table>` Table start and end
  - `<tr> </tr>` Table row start and end
  - `<th> </th>` Table header element
  - `<td> </td>` Table data element

Item	Price
Soy Milk	$4.39
Frosted Flakes	$3.89

```
<table border="1" summary="Prices">
 <tr>
     <th>Item</th><th>Price</th>
 </tr>
 <tr>
     <td>Soy Milk</td><td>$4.39</td>
 </tr>
 <tr>
   <td>Frosted Flakes</td><td>$3.89</td>
 </tr>
</table>
```

 11

Nested for Example

```
<title>Subtraction Table</title>
<script type = "text/javascript">
   document.writeln("<h3>Subtraction Table (X-Y)<\/h3>"
     + "<table summary=\"Subtraction\" border=\"1\">"
     + "<tr><th>Y\\X<\/th>")
   for(var i = 0; i <= 3; i++)
       document.write( "<th>" + i + "<\/th>");
   document.writeln("</tr>");
   for(i = 0; i <= 2; i++)
   {
     document.write("<tr><th>" + i + "</th>");
     for(var j = 0; j <= 3; j++)
     {
       document.write( "<td>" + (j - i) + "</td>");
     }
     document.writeln("</tr>");
   }
   document.writeln("</table>");
</script>
```

Subtraction Table (X-Y)

Y\X	0	1	2	3
0	0	1	2	3
1	-1	0	1	2
2	-2	-1	0	1

switch - case Structure

- A selection structure that can be used to select one of many branches

```
switch (Grade)
{
        case "P": case "p":
                window.alert("You Passed");
                break;
        case "F": case "f":
                window.alert("You Failed");
                break;
        default:
                window.alert("Incorrect letter Grade.");
}
```

 13

```
<head>    <title>Filtered Data Entry</title><script type="text/javascript">
    var Entry, Count = 0;
    do
    {
     Entry = window.prompt( "Enter the letter Grade or Q=Quit", "A B C D F Q");
     switch (Entry)
     {
       case "A": case "a":
         window.alert("Excellent work");
         break;
       case "B": case "b":
         window.alert("Good work");
         break;
       case "C": case "c":
         window.alert("Average work");
         break;
       case "D": case "d":
       case "F": case "f":
         window.alert("Poor work");
         break;
       case "Q": case "q":
         break;
       default:     // catch all other characters
         window.alert("Incorrect letter Grade.");
     }
    } while ( Entry !="Q" && Entry != "q");
    window.alert("Quit");   </script>  </head> <body>Click Refresh to repeat </body>
```

Exercise

- Create a program to display a multiplication table for two entered values designated by columns and rows
- For example if you enter 10 columns and 5 rows the following table would be generated

Multiplication Table (Row x Col)

C\R	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50

 15