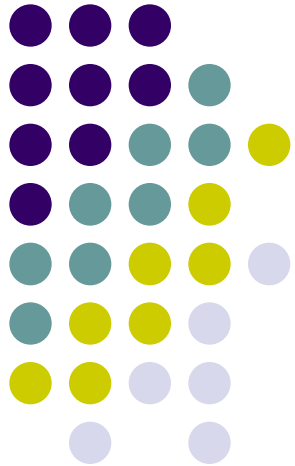
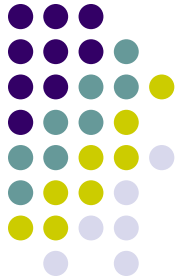
Loops



A C P K Siriwardhana,
BSc, MSc

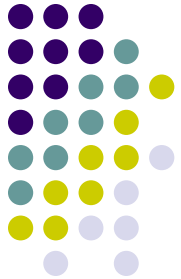
The for Loop Repetition Structure



- The **for** loop handles details of the counter-controlled loop "automatically".
- The initialization of the the loop control variable, the termination condition test, and control variable modification are handled in the **for** loop structure.

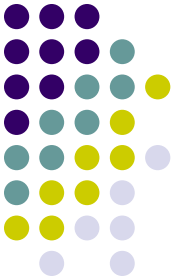
```
for ( i = 1; i < 101; i = i + 1)
{
initialization   modification
}          test
```

When Does a for Loop Initialize, Test and Modify?



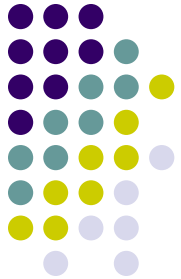
- a for loop
  - initializes the loop control variable before beginning the first loop iteration,
  - modifies the loop control variable at the very end of each iteration of the loop, and
  - performs the loop termination test before each iteration of the loop.
- The for loop is easier to write and read for counter-controlled loops.

A for Loop That Counts From 0 to 9



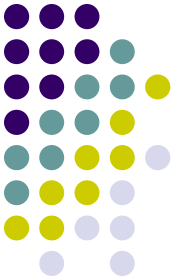
```
for(i = 0; i < 10; i = i + 1)
{
  alert("i is " + i);
}
```

We Can Count Backwards, Too



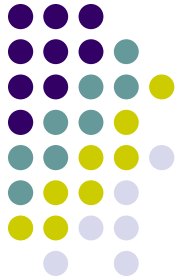
```
for(i = 9; i >= 0; i = i - 1)
{
  alert("i is " + i);
}
```

We Can Count By 2's ... or 7's ... or Whatever



```
for(i = 0; i < 10; i = i + 2)
{
  alert("i is " + i);
}
```

Nested Loops



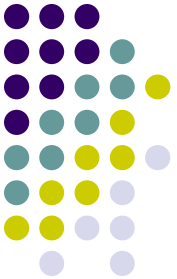
- Loops may be **nested** (**embedded**) inside of each other.
- Actually, any control structure (sequence, selection, or repetition) may be nested inside of any other control structure.
- It is common to see nested for loops.

Nested for Loops

```
1.  for (i = 1; i < 5; i = i + 1)
2.  {
3.    for( j = 1; j < 3; j = j + 1)
4.    {



5.    }
6.  }
```



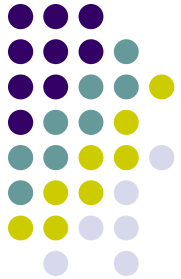
Nested for Loops

```
1.  for (i = 1; i < 5; i = i + 1)
2.  {
3.    for( j = 1; j < 3; j = j + 1)
4.    {
5.      if (j % 2 == 0)
6.      {
7.        document.write("O");
8.      }
9.      else
10.     {
11.       document.write("X");
12.     }
13.   }
14.   document.write("<br />");
15. }
```

How many times is the "if" statement executed?

What is the output ?

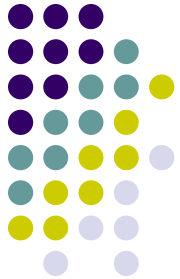
The break **Statement**



- The **break** statement can be used in **while**, **do-while**, and **for** loops to cause premature exit of the loop.

- THIS IS ***NOT*** A RECOMMENDED CODING TECHNIQUE.

Example break in a for Loop



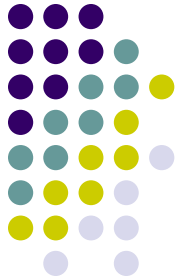
```
var i;
for(i = 1; i < 10; i = i + 1)
{
  if(i == 5)
  {
    break;
  }
  document.write(i + " ");
}
document.write("Broke out of loop at i = " + i);
```

OUTPUT:

1 2 3 4

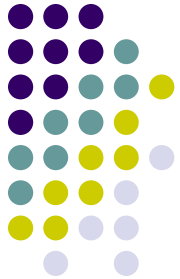
Broke out of loop at i = 5.

The continue **Statement**



- The **continue** statement can be used in **while**, **do-while**, and **for** loops.
- It causes the remaining statements in the body of the loop to be skipped for the current iteration of the loop.
- THIS IS ***NOT*** A RECOMMENDED CODING TECHNIQUE.

Example continue in a for Loop



```
var i;
for(i = 1; i < 10; i = i + 1)
{
  if(i == 5)
  {
    continue;
  }
  document.write(i + " ");
}
document.write("Done.");
```

OUTPUT:

1 2 3 4 6 7 8 9

Done.