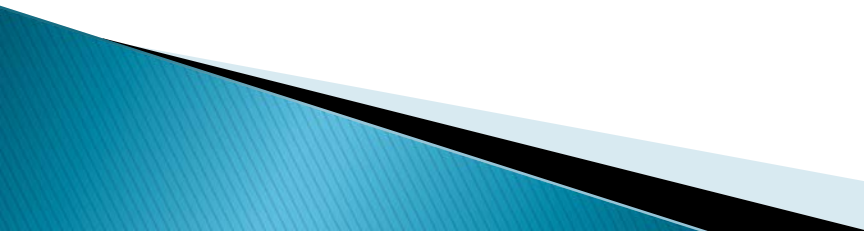


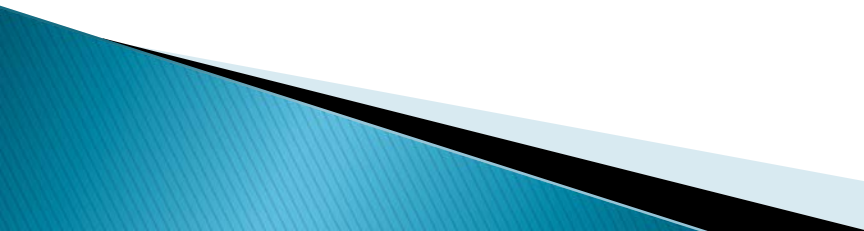
Lecture 4

Switch Statement

The switch Statement

- ▶ The *switch statement* provides another way to decide which statement to execute next
 - ▶ The *switch* statement evaluates an expression, then attempts to match the result to one of several possible *cases*
 - ▶ Each case contains a value and a list of statements
 - ▶ The flow of control transfers to statement associated with the first case value that matches
- 

The switch Statement

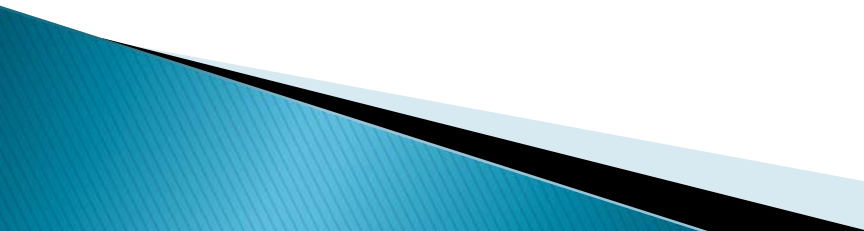
- ▶ Often a *break statement* is used as the last statement in each case's statement list
 - ▶ A *break* statement causes control to transfer to the end of the *switch* statement
 - ▶ If a *break* statement is not used, the flow of control will continue into the next case
 - ▶ Sometimes this may be appropriate, but often we want to execute only the statements associated with one case
- 

The switch Statement

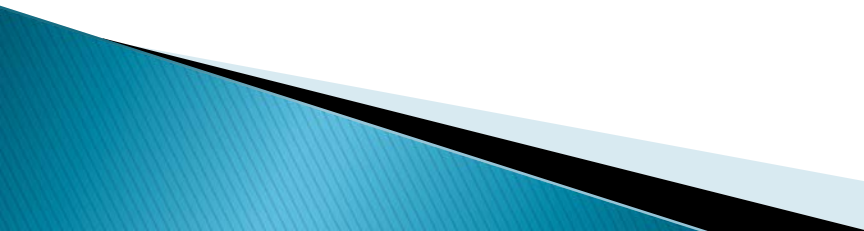
- ▶ An example of a switch statement:

```
switch (option)
{
    case 'A':
        aCount++;
        break;
    case 'B':
        bCount++;
        break;
    case 'C':
        cCount++;
        break;
    default:
        otherCount++;
        break;
}
```

The switch Statement

- ▶ A `switch` statement can have an optional *default case*
 - ▶ The default case has no associated value and simply uses the reserved word `default`
 - ▶ If the default case is present, control will transfer to it if no other case value matches
 - ▶ If there is no default case, and no other value matches, control falls through to the statement after the switch
- 

The switch Statement

- ▶ The expression of a `switch` statement must result in an *integral type*, meaning an integer (byte, short, int,) or a char
 - ▶ It cannot be a floating point value (float or double)
 - ▶ The implicit test condition in a `switch` statement is equality
 - ▶ You cannot perform relational checks with a `switch` statement
- 

The switch Statement

- ▶ The general syntax of a switch statement is:

```
switch ( expression )  
{  
  case value1 :  
    statement-list1  
  case value2 :  
    statement-list2  
  case value3 :  
    statement-list3  
  case ...  
}
```

switch
and
case
are
reserved
words



If *expression*
matches *value2*,
control jumps
to here