Overview (1):

- Before We Begin
  - Some administrative details
  - Some questions to consider
- The “If” Statement
  - Introduction to the If statement
  - The If/Else statement
  - The nested If statement
  - Examples
    - Exercise 4-2

Before We Begin
Administrative Details (1):

- Lab Exercise 3-3
  - Exercise has been graded and will be distributed back to you after today’s lecture
  - Has been graded /1 (e.g., either it is correct or not correct)

- Reminder
  - You should be working on Ex 4-4 this week
  - Test 1 will be held February 8 2006
    - More details to follow on Friday

Some Questions to Consider (1):

- What is a Boolean expression?
- What is a comparison operator?
- List some of the common comparison operators
- How do we use a comparison operator (and how many values (variables) does it require)?
- Can we compare two variables of different types?

If Statements
Introduction (1):
- So Far, Boolean Expressions → Limited Use
  - We can obtain a Boolean value by evaluating the Boolean expression but we have not used it!
  - Basically, as hinted at in the previous lecture, the Boolean expression provides one of two values → True or False
  - What about executing one set of statements if the value is True and another if it is False?
  - This is actually the most common use of a Boolean condition → directly used in the If statement

Introduction (2):
- The If Statement
  - Select whether a set of statements are executed
  - Simplest type of If statement → block If statement
  - Syntax of the block If statement
    - Other statements in sub-program
    - If (Boolean Value) Then statement 1 statement 2 ...
    - End If
    - More statements

Introduction (3):
- The If Statement (cont.)
  - An everyday analogy to the If statement
    - Enter shop
    - Look around
    - Find item
    - If (itemPrice < myCash) Then Buy item
    - End If
    - Leave shop
      - "Statements" before the If statement that are executed in sequence before the IF
      - If it is true that itemPrice is less than myCash we "execute" the statement "Buy item" otherwise we do not
      - Regardless, the statement "Leave shop" is always executed
## Introduction (4):

### The Else Statement

- As we previously saw, we can generally use a Boolean value (expression) to decide one of two sets of statements to execute.

<table>
<thead>
<tr>
<th>Boolean Value</th>
<th>Boolean expression that is evaluated to a single value (True or False) or simply a Boolean variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other statements in sub-program</td>
<td></td>
</tr>
<tr>
<td>If (Boolean Value) Then statements</td>
<td></td>
</tr>
<tr>
<td>Else statements</td>
<td></td>
</tr>
<tr>
<td>End If</td>
<td></td>
</tr>
<tr>
<td>More statements</td>
<td></td>
</tr>
</tbody>
</table>

"Else" statement must be on a new line by itself.

## Introduction (5):

### The ElseIf Statement

- So far, depending on the value of the Boolean expression:
  - One set of statements is executed if True
  - One set of statements is executed if False
- We can actually expand on this and allow for choosing between one of three sets of statements to be executed:
  - This actually is extremely useful!

## Introduction (6):

### The ElseIf Statement

**ElseIf syntax**

<table>
<thead>
<tr>
<th>Boolean Value</th>
<th>Boolean expression that is evaluated to a single value (True or False) or simply a Boolean variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other statements in sub-program</td>
<td></td>
</tr>
<tr>
<td>If (Boolean expression 1) Then statement(s)</td>
<td></td>
</tr>
<tr>
<td>ElseIf (Boolean expression 2) Then statement(s)</td>
<td></td>
</tr>
<tr>
<td>Else statement(s)</td>
<td></td>
</tr>
<tr>
<td>End If</td>
<td></td>
</tr>
<tr>
<td>More statements</td>
<td></td>
</tr>
</tbody>
</table>

If (Boolean Value 1) = True then execute the statements within the If block otherwise if (Boolean value 2) = True then execute statements in "ElseIf" block otherwise, if both Boolean values are false then execute statements in the Else block.

As soon as one true expression is encountered and corresponding code is executed, then "exit" the nested if statement.
Introduction (7):

- General ElseIf Statement (Nested IF)
  - Multiple ElseIf statements
  - Other statements in sub-program
  - If (Boolean expression 1) Then
    statement(s)
  - ElseIf (Boolean expression 2) Then
    statement(s)
  - ... (More ElseIf statements)
  - Else
    statement(s)
  - End If
  - More statements
  - The multiple ElseIf statements as well as the Else statement are all optional

Exercise 4-2 (1):

- If Statement
  - Examples
    - Ex 4-2: The number guessing game
      - Simple game program whereby the player (user) tries to guess a number that has been selected randomly between 1 and 100

Exercise 4-2 (2):

- Exercise 4-2 Brief Description
  - User enters a guess and presses the "Check It" button
  - Message is displayed indicating whether guess is too small, too large or equal to some random number chosen by the program
  - When the "Play Again" button is pressed, we start game over
    - New random number is chosen
Exercise 4-2 (3):

Exercise 4-2 Brief Description (cont.)

- But how do we choose a random number?
  - Most programming languages provide one or more functions that generate random numbers
  - Visual Basic has the "Rnd" function, which generates a random number between 0 and 1

```vba
Dim rndValue As Single
rndValue = Rnd()
```

Exercise 4-2 (4):

Exercise 4-2 Brief Description (cont.)

- But if random number is only between 0 and 1, how does this help us with our game that allows numbers between 0 and 100 or in any other range other than 0 and 1?
  - Given a random number between 0 and 1, we can accommodate any range!
  - We simply multiply the randomly generated number by the maximum of our required range
  - But what if the minimum value of our range is greater than 0? How do we account for this?