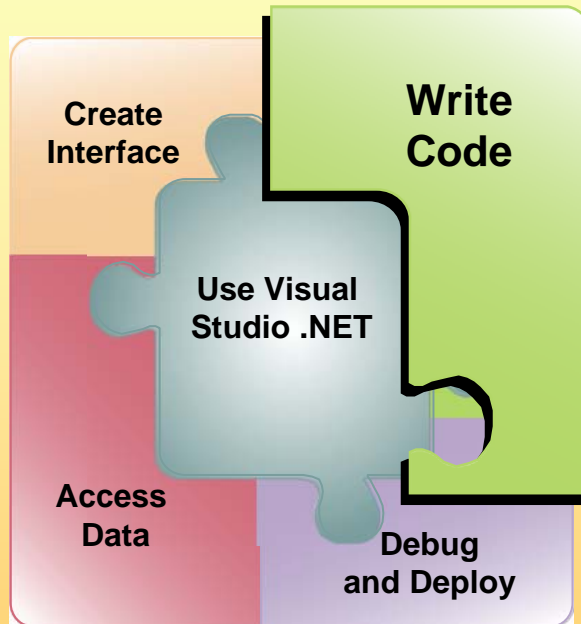


# **Module 5: Decision Structures and Loops**

# Overview



- Using Conditional Expressions
- Using Decision Structures
- Using Conditional Loop Structures

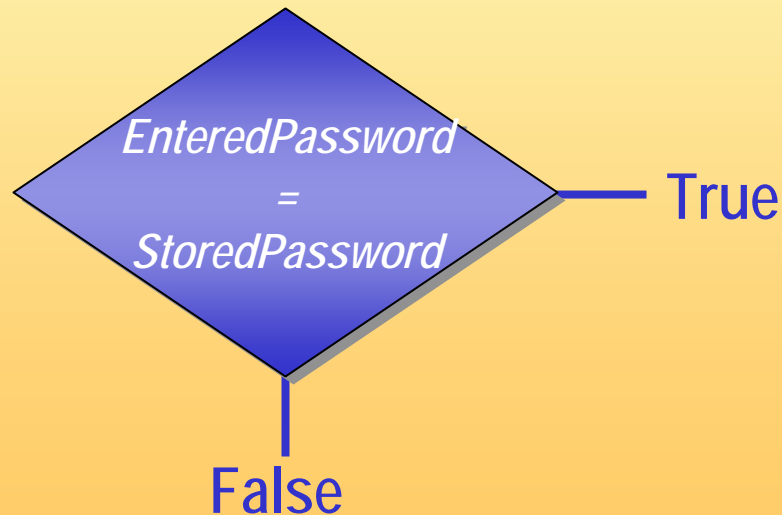
# Lesson: Using Conditional Expressions

- What Are Conditional Expressions?
- How to Use Arithmetic Operators
- How to Use Comparison Operators
- How to Use Logical Operators
- How to Combine Comparison and Logical Operators

# What Are Conditional Expressions?

- Conditional expressions:

- Include a condition to be tested as **True** or **False**
- Include an operator to specify what the test of the condition is



If the password *is* the correct password, the condition is True

# How to Use Arithmetic Operators

- Symbols that evaluate conditional expressions
- Can perform arithmetic operations
- Syntax:

```
expression1 arithmetic operator expression2
```

- Example:

```
Dim x As Integer  
x = 52 * 17  
x = 120 / 4  
x = 67 + 34  
x = 32 - 12  
x = 23 ^ 3
```

# How to Use Comparison Operators

- Symbols that evaluate conditional expressions and return a Boolean value
- Can compare numbers or strings
- Syntax:

```
expression1 comparison operator expression2
```

- Example:

```
Dim Quantity As Integer  
Dim LargeOrder As Boolean  
LargeOrder = Quantity > 1000
```

# How to Use Logical Operators

- Logical operators perform logical evaluation of expressions and return a Boolean value
- Syntax:

```
expression1 logical operator expression2
```

- Example:

```
OrderedLastMonth And OrderDelivered
```

# How to Combine Comparison and Logical Operators

- You can combine comparison operators and logical operators with conditional statements
- Example:

Comparison Operators

Logical Operator

```
LateActiveCustomer = DaysOverDue >= 60 And ActiveCustomer
```

# Practice: Evaluating Conditional Expressions

- Use the sample application to calculate the results of these expressions:

TestString = TestString	0 And 0
TestString = Teststring	-1 And 0
TestString < TestString	-1 And -1
Test < TestString	-1 Or -1
100 > 10	-1 Xor -1
10 < 10	-1 Xor 0
10 <= 10	0 Xor 0



# Lesson: Using Decision Structures

## ■ If...Then

```
If Sales > 10000 Then  
    Bonus = .10 * Sales  
End If
```

## ■ If...Then...Else

```
If Sales > 10000 Then  
    Bonus = .10 * Sales  
Else  
    Bonus = 0  
End If
```

## ■ If...Then...ElseIf

```
If Sales > 10000 Then  
    Bonus = .10 * Sales  
ElseIf Sales > 5000 Then  
    Bonus = .05 * Sales  
Else  
    Bonus = .02 * Sales  
End If
```

## ■ Select Case

```
Select Case Rank  
    Case 1  
        Bonus = 0  
    Case 2,3  
        Bonus = .05 * Sales  
    Case 4 to 6  
        Bonus = .10 * Sales  
    Case Else  
        Bonus = .15 * Sales  
End Select
```

# How to Use If...Then Statements

- Use for a decision that is either True or False
- If the condition is True, the statements following the If statement are executed
- If the condition is False, the statements following the If statement are not executed

```
If Sales > 10000 Then  
    Bonus = .10 * Sales  
End If
```

# How to Use If...Then...Else Statements

- Use for a decision with at least two options
- Each If statement must have corresponding End If
- If the condition is True, the statements following the If statement are executed
- If the condition is False, the statements following the If statement are not executed

```
If Sales > 10000 Then
    Bonus = .10 * Sales
Else
    Bonus = 0
End If
```

# How to Use If...Then...Else Statements

- Use to nest decision statements
- Each If statement must have corresponding End If
- Elseif statements do not have their own End If
- Elseif statements cannot appear after an Else
- If the condition is True, the statements following the If statement are executed

```
If Sales > 10000 Then  
    Bonus = .10 * Sales  
ElseIf Sales > 5000 Then  
    Bonus = .05 * Sales  
Else  
    Bonus = .02 * Sales  
End If
```

# How to Use Select Case Statements

- Select a block of code to execute based on a list of possible choices
- Use as an alternative to complex nested If...Then...Else statements
- If multiple Case statements are true, only the statements belonging to the first true Case statement are executed

## Select Case Rank

### Case 1

Bonus = 0

### Case 2,3

Bonus = .05 \* Sales

### Case 4 to 6

Bonus = .10 \* Sales

### Case Else

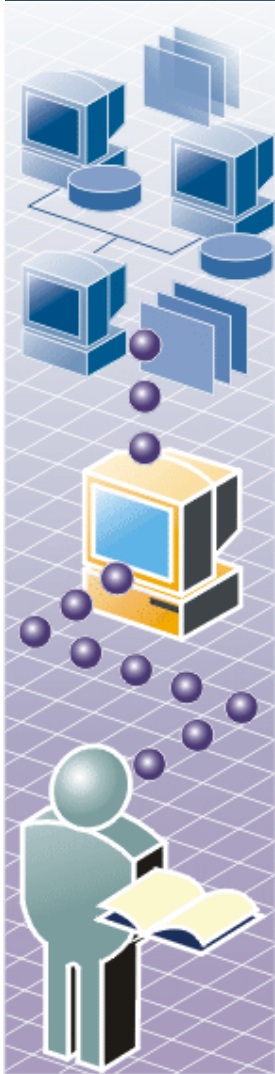
Bonus = .15 \* Sales

## End Select

# Guidelines for Choosing a Decision Structure

- Use If...Then statements to control the execution of a single block of code
- Use If...Then...Else statements to control the execution of two mutually exclusive sections of code
- Use Select Case statements when you have a list of possible values

# Lab 5.1: Using Decision Structures



- Exercise 1: Checking User Input
- Exercise 2: Confirming Application Close

# Lesson: Using Conditional Loop Structures

- How to Use For...Next Statements
- How to Use For Each...Next Statements
- How to Use Do...Loop Statements
- How to Use Exit Statements

# How to Use For...Next Statements

- Use when you know the number of times you want to repeat execution of the code

```
For NamePos = 0 to 4  
    MessageBox.Show(Names(NamePos))
```

```
Next
```

```
' In reverse order
```

```
For NamePos = 4 to 0 Step -1  
    MessageBox.Show(Names(NamePos))
```

```
Next
```

# How to Use For Each...Next Statements

- A collection is a set of objects that are grouped together and referred to as a unit. For example:
  - Items in a list box are part of an **Items** collection
  - Every form has a **Controls** collection that represents all the controls on that form
- For Each ... Next statements are used to loop through elements in a collection

```
Sub LightBlueBackground (. . .)
    Dim ThisControl As System.Windows.Forms.Control
    For Each ThisControl In ThisForm.Controls
        ThisControl.BackColor = System.Drawing.Color.LightBlue
    Next ThisControl
End Sub
```

# How to Use Do...Loop Statements

## ■ Do...Loop Until

- Runs the code in the loop and then evaluates the condition. Repeats until the condition evaluates as **True**.

## ■ Do Until...Loop

- Runs code in the loop only if the condition evaluates as **False**, and repeats until the test expression tests as **True**.

## ■ Do...Loop While

- Runs the code in the loop and then evaluates the condition. Repeats until the condition evaluates as **False**.

## ■ Do While...Loop

- Runs the code in the loop only if the condition evaluates as **True**, and repeats until the test expression tests as **False**.

# How to Use Exit Statements

- Use to leave Do loops or For loops early when a condition is met

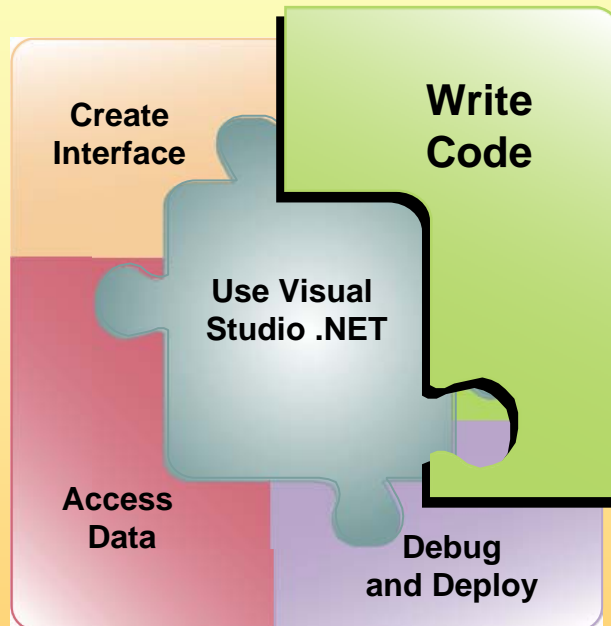
```
Do Until y = -1  
  If x < 0 Then Exit Do  
    x = Sqrt (x)  
  If y > 0 Then Exit Do  
    y = y + 3  
  If z = 0 Then Exit Do  
    z = x / y  
Loop
```

# Practice: Designing Loop Structures



- In each of four scenarios, decide which loop structure you would use to solve a problem
- In the development environment, use For...Next statements to create a multiplication table

# Review



- Using Conditional Expressions
- Using Decision Structures
- Using Conditional Loop Structures