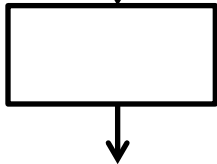
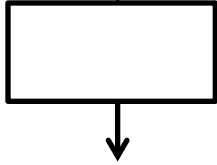
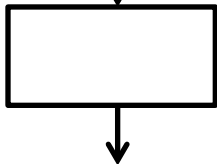
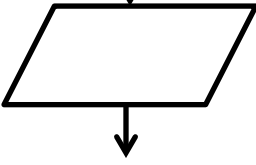


# CIS1400 -- Programming Logic and Technique

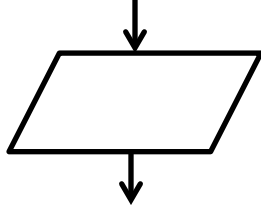
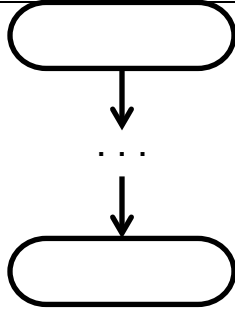
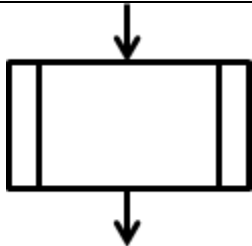
## Statement Conversions

### Sequential Statements

Statement	Pseudocode	Flowchart	Visual Basic
Variable Declaration	Declare Integer <i>varName</i> Declare Real <i>varName</i> Declare String <i>varName</i> Declare <i>Boolean</i> <i>varName</i>		Dim <i>varName</i> As Integer Dim <i>varName</i> As Double Dim <i>varName</i> As String Dim <i>varName</i> As Boolean
Constant Declaration	Constant Integer <i>constName</i> = <i>value</i> Constant Real <i>constName</i> = <i>value</i> Constant String <i>constName</i> = " <i>value</i> " Constant Boolean <i>constName</i> = True or False		Const <i>constName</i> As Integer = <i>value</i> Const <i>constName</i> As Double = <i>value</i> Const <i>constName</i> As String = " <i>value</i> " Const <i>constName</i> As Boolean = True or False
Assignment/Calculation	Set <i>varName</i> = <i>expression</i>		<i>varName</i> = <i>expression</i>
Input	Input <i>varName</i>		<i>varName</i> = CInt(Console.ReadLine()) <i>varName</i> = Cdbl(Console.ReadLine()) <i>varName</i> = Console.ReadLine()

# CIS1400 -- Programming Logic and Technique

## Statement Conversions

Statement	Pseudocode	Flowchart	Visual Basic
Output	Display "Enter a value: " Display "Your variable is ", varName		Console.Write("Please input a value: ") Console.WriteLine("Your variable is " & varName)
Module Definition	Module ModName*(dataType Ref** parmName) ... End Module		Sub SubName*(ByVal ByRef parmName AS dataType) ... End Sub
Module Call	Call SubName(argVal)		SubName(argVal)

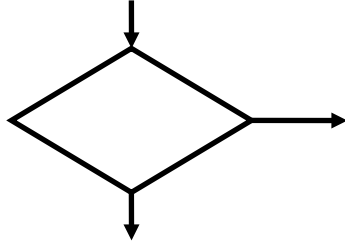
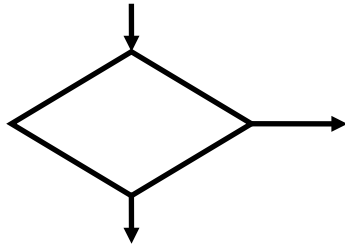
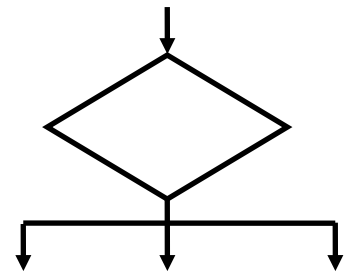
\* multiple arguments/parameters separated by commas

\*\* Ref keyword only needed for pass by reference parameters

# CIS1400 -- Programming Logic and Technique

## Statement Conversions

### Selection Statements

Statement	Pseudocode	Flowchart	Visual Basic
Single Alternative	<pre> If <i>condition</i> Then     <i>StatementsIfTrue</i> End If </pre>		<pre> If <i>condition</i> Then     <i>StatementsIfTrue</i> End If </pre>
Dual Alternative	<pre> If <i>condition</i> Then     <i>StatementsIfTrue</i> Else     <i>StatementsIfFalse</i> End If </pre>		<pre> If <i>condition</i> Then     <i>StatementsIfTrue</i> Else     <i>StatementsIfFalse</i> End If </pre>
Multiple Alternative	<pre> Select <i>testExpr</i>     Case Val1:         <i>StatementsIfVal1</i>     Case Val2:         <i>StatementsIfVal2</i>     Default:         <i>StatementsIfDefault</i> End Select </pre>		<pre> Select Case <i>testExpr</i>     Case <i>Expression1</i>         <i>StatementsIfExpression1</i>     Case <i>Expression2</i>         <i>StatementsIfExpression2</i>     Case Else         <i>StatementsIfElse</i> End Select </pre>