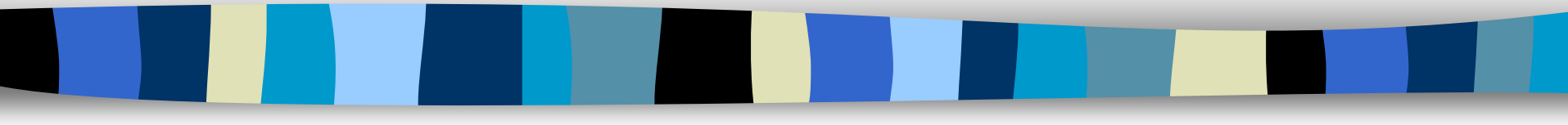
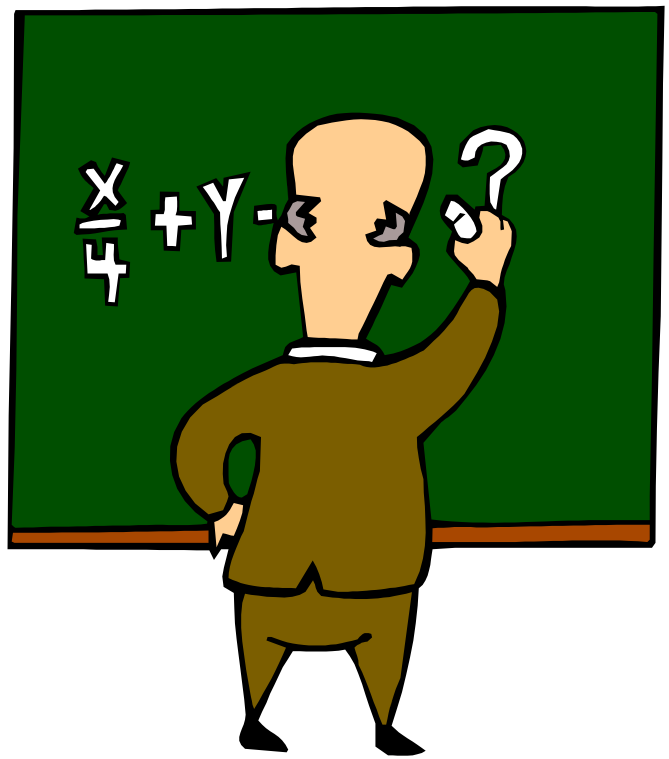
Let’s Get Visual!



Learn about variables… what they are and how to use them.

What Are Variables?

- A variable is a word or symbol that represents a value to the computer.
- Just like in Logo, you can call a variable anything, and set its value to be a number or a string of one or more words.



Declaring a Variable in VB

- To tell VB to make a variable “num” that holds a whole number (an integer), you say:

Dim num As Integer

- To tell VB to make a variable “sentence” that holds a string of word(s) (a string), you say:

Dim sentence As String

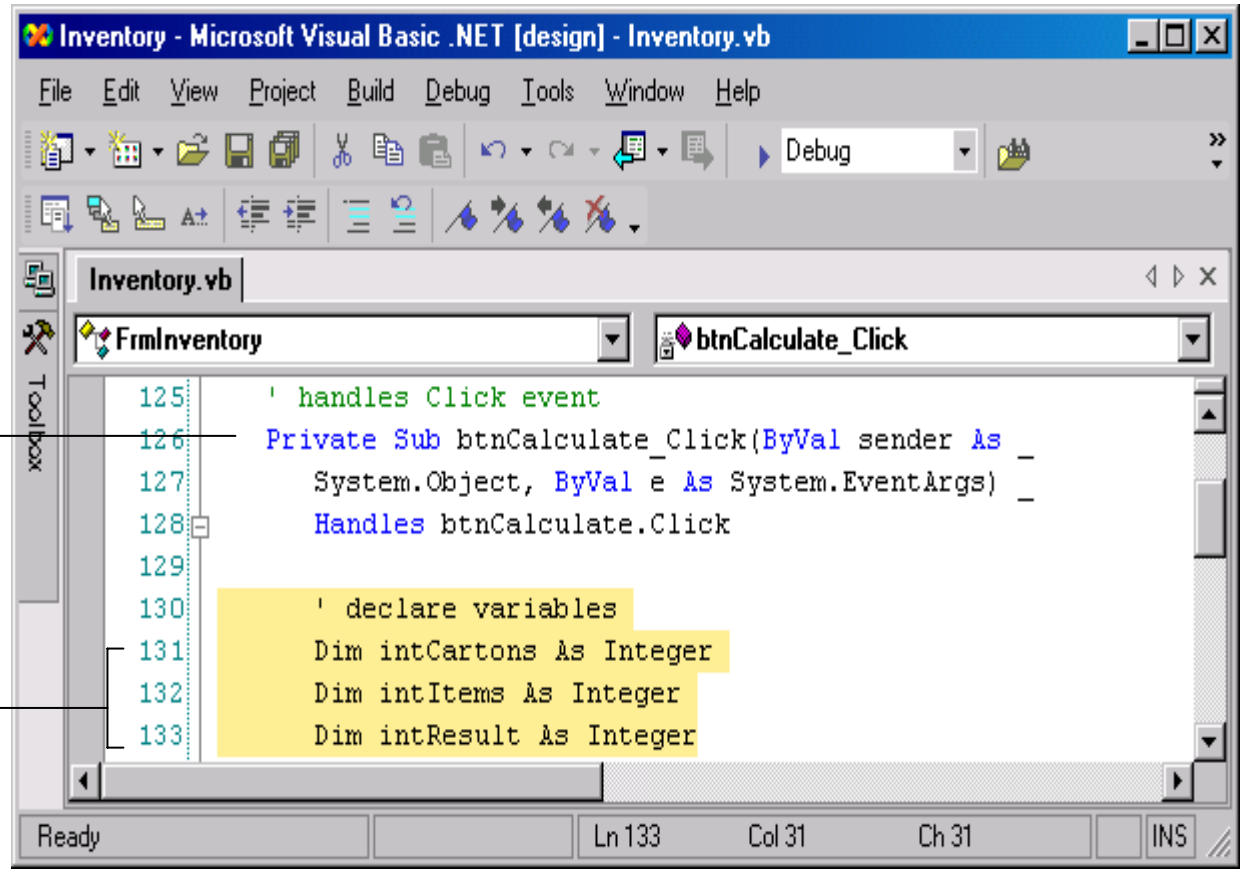
- To tell VB to make a variable “deci” that holds a decimal number (a floating point number), you say:

Dim deci As Float

For Example…

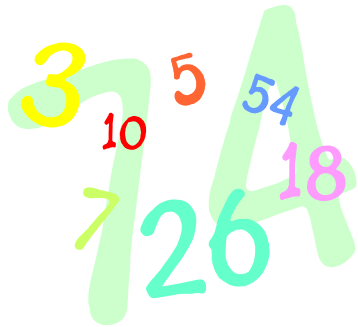
Click event handler

Variable declarations



Giving a Variable a Value

So we've declared three variables, but they don't have values yet. Here's a couple ways you can put values into variables:

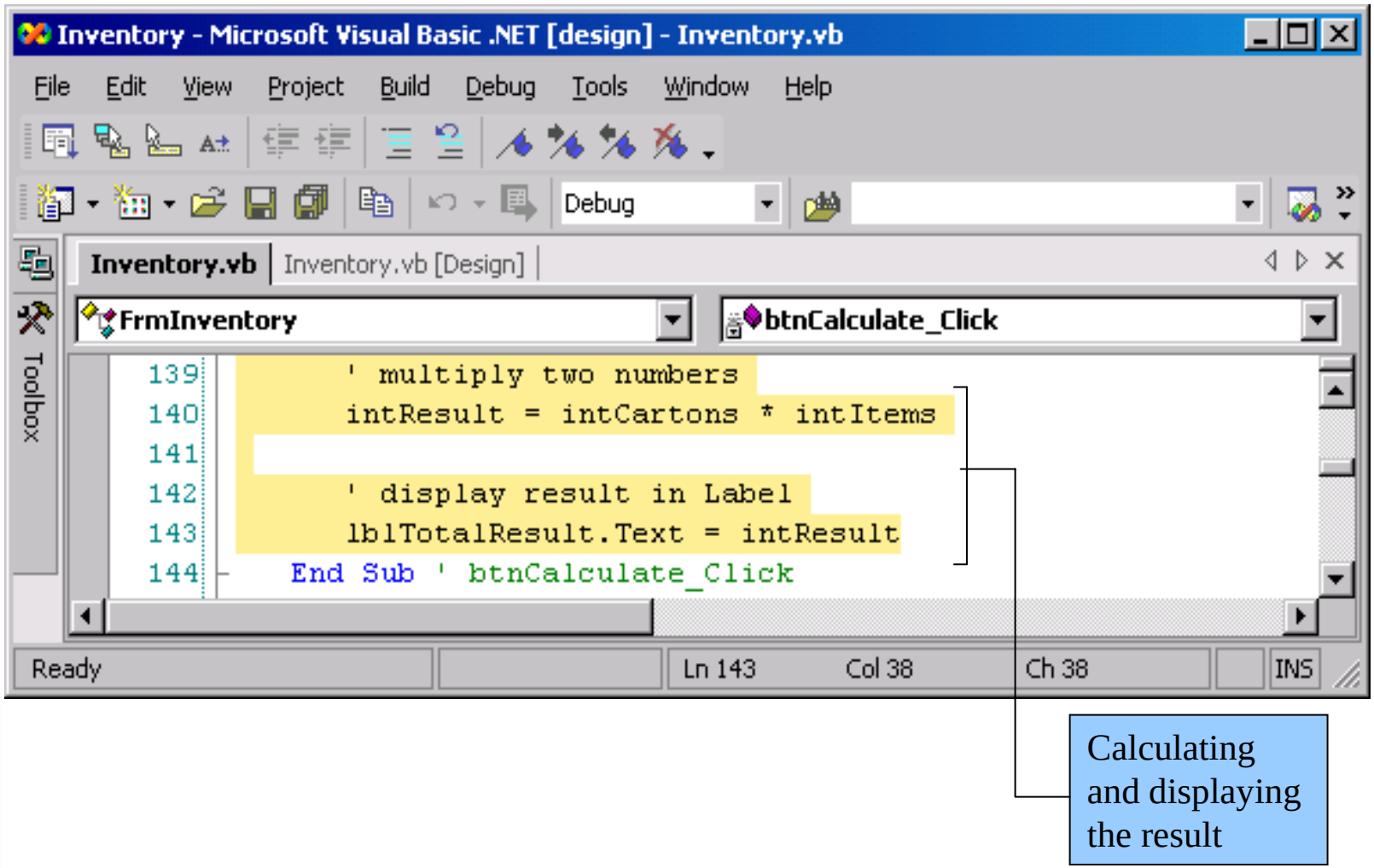


- Set the value of the variable to be the same as the contents of a text box.
- Set the value of the variable equal to any number.
- Set the value of the variable equal to the sum, product, difference, etc. of any other numbers or text box values.

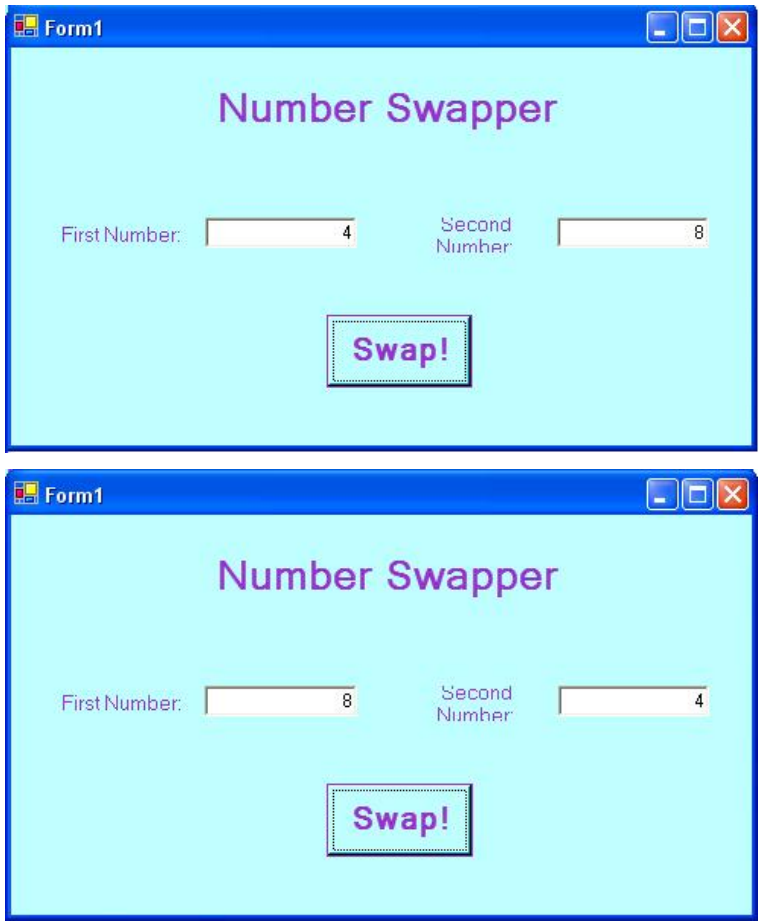
When Would You Need A Variable?

- When you need to store a value when going from one step of a program to another.
- When there's a value you want to use over and over.
- Can you think of anymore?

Variable Example



Assignment: Number Swapper



- Create the GUI at left.
- Double-click the button to open the code.
- Declare a variable to hold intermediate values.
- Use that variable to write the code to swap the values in the text boxes.

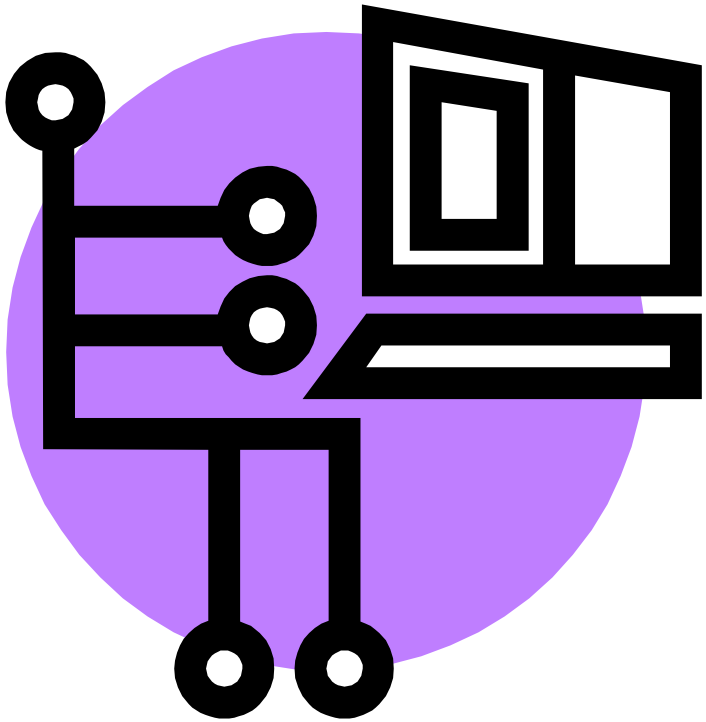
Let’s Get Visual!

Learn what a conditional statement is and how to use it.

Conditional Statements

- There are tons of real-life situations where you will do something in one scenario and do nothing otherwise.
- How many of these situations can you think of?

Conditional Statements



- There are a million situations in programming when you'll want the computer to behave this way.
- Programming languages like VB have an if-then conditional statement built-in so that you can communicate the situation to the computer.

Syntax: If Then Statement

Syntax

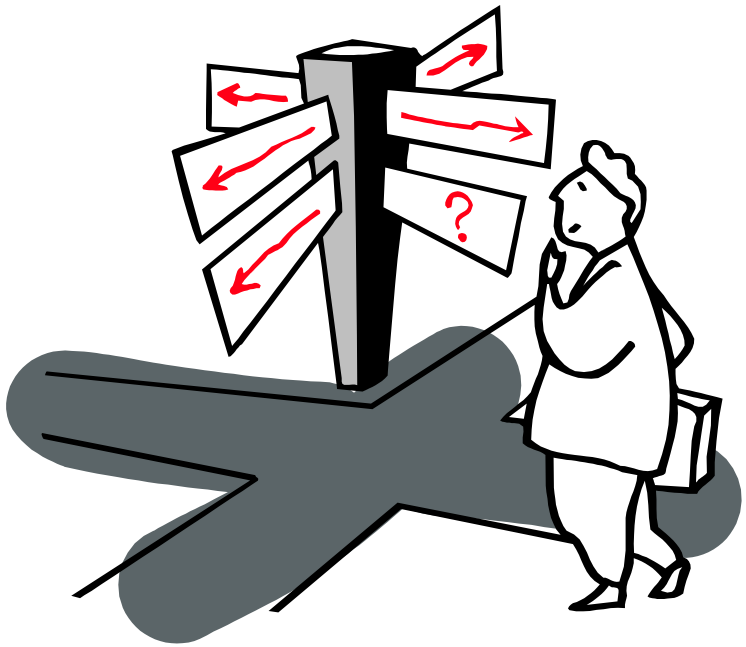
```
If condition Then
    [ statements ]
End If
```

Figure 7.5 `If...Then` statement syntax.

```
If intStudentGrade >= 90 Then
    lblDisplay.Text = "A"

End If
```

Conditional Statements



- More often, you will perform one action in one scenario and a completely different action otherwise.
- For this situation, VB has an if-then-else statement to communicate the situation to the computer.

Syntax: If Then Else Statement

Syntax

`If` *condition* `Then`
  [*statements*]
`Else`
  [*statements*]
`End If`

Figure 7.7 `If…Then Else` statement syntax.

```
If intStudentGrade >= 90 Then
   lblDisplay.Text = "A"
Else
   If intStudentGrade >= 80 Then
      lblDisplay.Text = "B"
   Else
      If intStudentGrade >= 70 Then
         lblDisplay.Text = "C"
      Else
         If intStudentGrade >= 60 Then
            lblDisplay.Text = "D"
         Else
            lblDisplay.Text = "F"
         End If
      End If
   End If
End If
```

Syntax: If Then ElseIf Statement

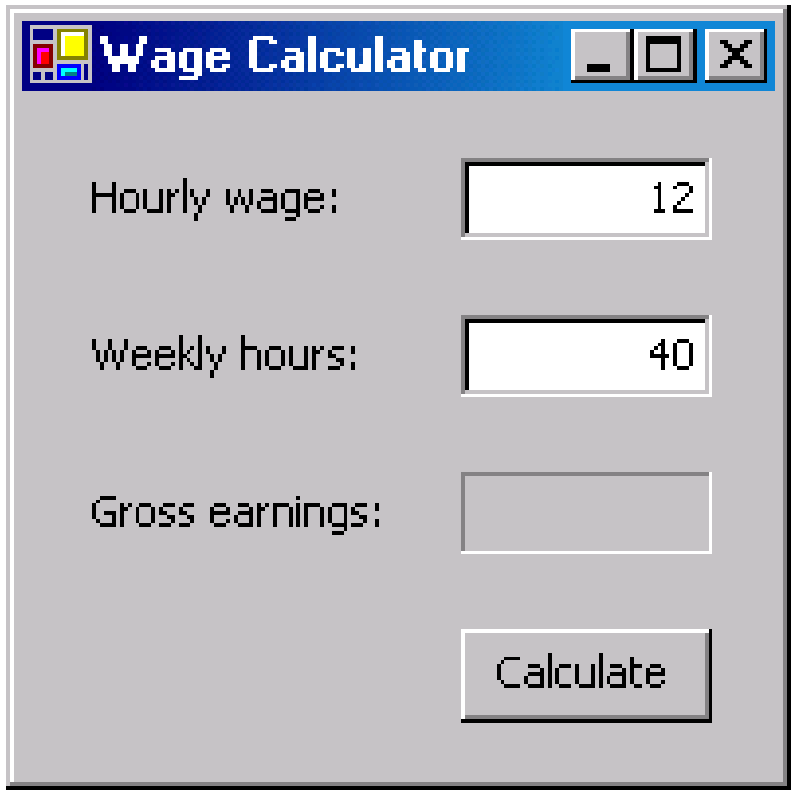
This structure is a fancy if-then-else statement that makes for much simpler code. It allows you to keep checking for true, but only until you find it, with very little code.

```
If intStudentGrade >= 90 Then
   lblDisplay.Text = "A"
ElseIf intStudentGrade >= 80 Then
   lblDisplay.Text = "B"
ElseIf intStudentGrade >= 70 Then
   lblDisplay.Text = "C"
ElseIf intStudentGrade >= 60 Then
   lblDisplay.Text = "D"
Else
   lblDisplay.Text = "F"
End If
```

Relational Operators

Conditional Operator	Code Example	Meaning
>	intX > intY	intX is greater than intY
<	intX < intY	intX is less than intY
≥	intX ≥ intY	intX is greater than or equal to intY
≤	intX ≤ intY	intX is less than or equal to intY
=	intX = intY	intX is equal to intY
≠	intX ≠ intY	intX is not equal to intY

Assignment: Wage Calculator



- Create the GUI at left.
- Double-click the button to open the code.
- Calculate the total wages earned.
- If the person worked more than 40 hours, s/he gets 1.5 times the wage per hour for that extra time.
- Use a conditional statement to check for this, and variables to store all the steps of the program.