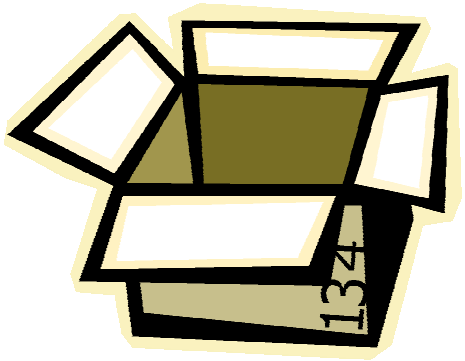
Arrays

- Creating and Accessing Arrays
- Using Arrays
- Some Additional Types of Arrays

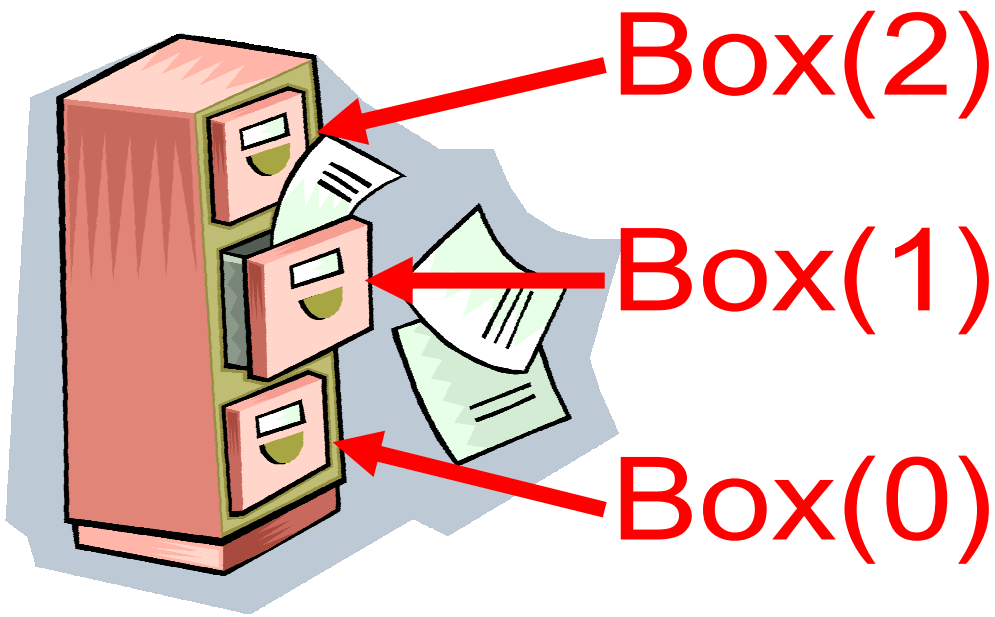
Variable vs Array

Simple Variable



Dim Box as Integer

Array Variable



Dim Box(2) as Integer

Simple and Array Variables

- A **variable** (or simple variable) is a name to which Visual Basic can assign a single value.
- An **array variable** is a collection of simple variables of the same type to which Visual Basic can efficiently assign a list of values.

Creating and Accessing Arrays

- Declaring an Array Variable
- The Load Event Procedure
- The GetUpperBound Method
- ReDim Statement
- Using an Array as a Frequency Table

Example

- Suppose that you want to evaluate the exam grades for 30 students and to display the names of the students whose scores are above average.

```
Private Sub btnDisplay_Click(...) _
                                Handles btnDisplay.Click
  Dim student1 As String, score1 As Double
  Dim student2 As String, score2 As Double
  Dim student3 As String, score3 As Double
```

It’s better when Using Arrays

Upper bound of subscripts
in the array

```
Dim student(29) As String
Dim score(29) As Double
```

Array name

Data type

Putting Values into an Array

student(0) = "Tom Brown"

Read: "student sub zero equals Tom Brown"

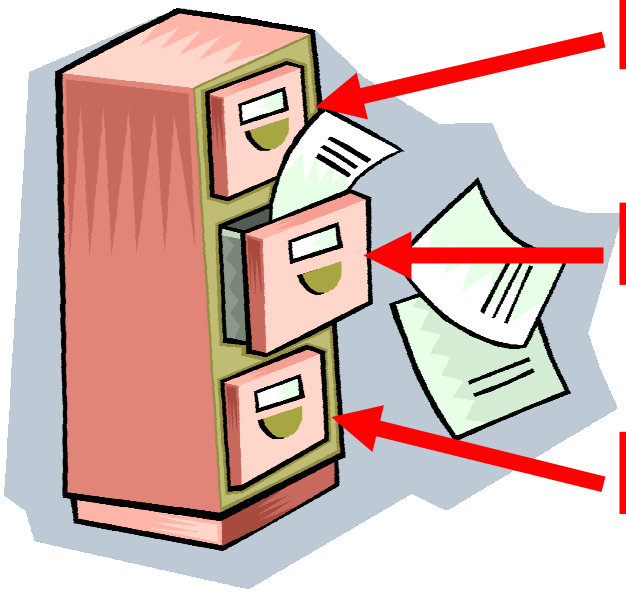
Which means that the string "Tom Brown" is being stored at the first location in the array called student… because all arrays begin counting at 0.

Array Terminology

- Dim arrayName(n) As DataType
- 0 is the "lower bound" of the array
- n is the "upper bound" of the array – the last available subscript in this array
- The number of elements, n + 1, is the *size* of the array

Upper Bound & Lower Bound

Array Variable



Box(2)

This is Upper Bond

Box(1)

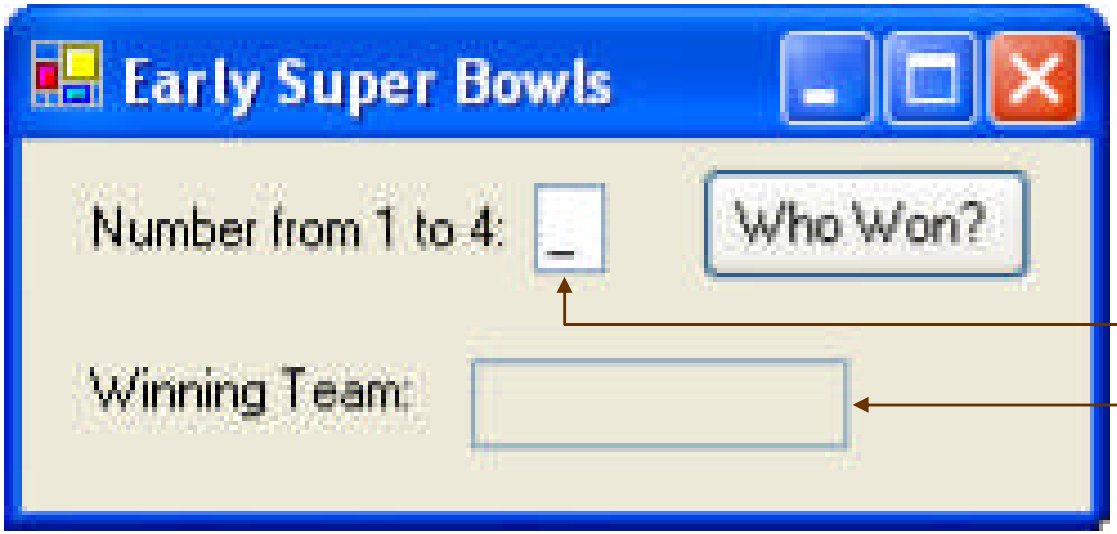
Box(0)

This is Lower Bond

Dim Box(2) as Integer

Number of elements in this array is **3**

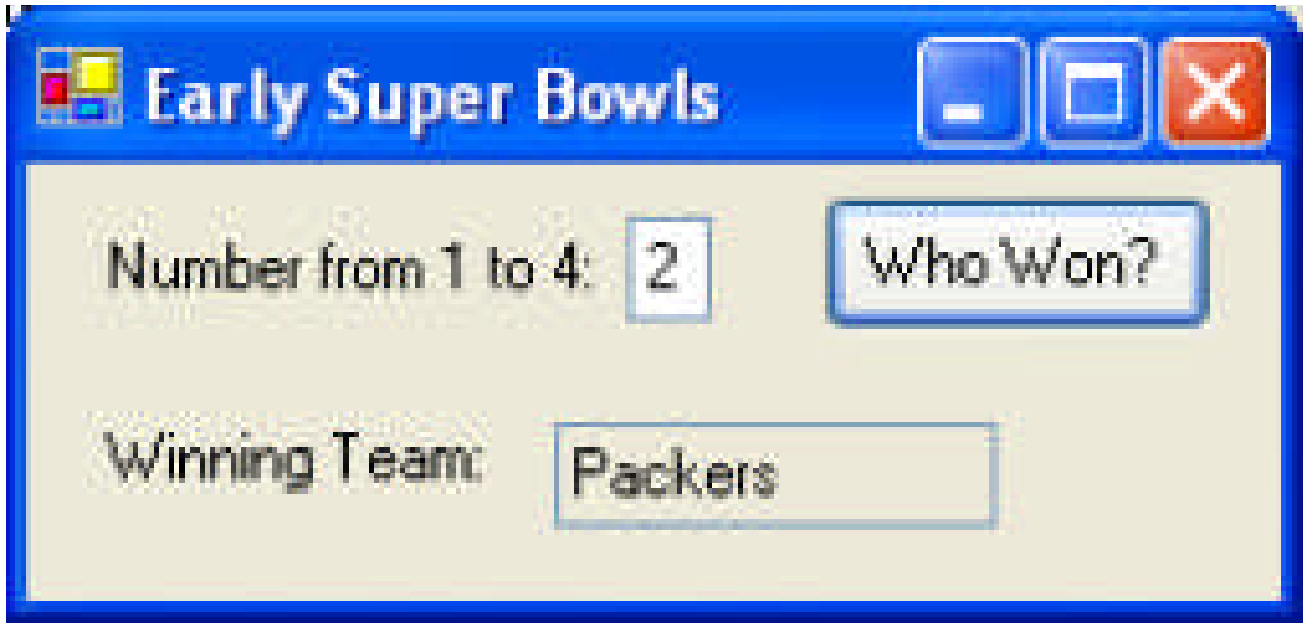
Lab sheet 7.1: Form



Example – Initialize the Array in the Button

```
Private Sub btnWhoWon_Click(...) _
                                Handles btnWhoWon.Click
  Dim teamName(3) As String
  Dim n As Integer
  'Place Super Bowl Winners into the array
  teamName(0) = "Packers"
  teamName(1) = "Packers"
  teamName(2) = "Jets"
  teamName(3) = "Chiefs"
  'Access array
  n = CInt(txtNumber.Text)
  txtWinner.Text = teamName(n - 1)
End Sub
```

Lab sheet 7.1: Early Super Bowls



Load Event Procedure

Occurs as the Form loads in memory

```
Private Sub frmName_Load(...) _
                    Handles MyBase.Load
```

The keyword MyBase refers to the form being loaded. This event procedure is a good place to assign values to an array.

Example- Initialize the Array in Form Load

```
Dim teamName(3) As String

Private Sub btnWhoWon_Click(...) Handles btnWhoWon.Click
  Dim n As Integer
  n = CInt(txtNumber.Text)
  txtWinner.Text = teamName(n - 1)
End Sub

Private Sub frmBowl_Load(...) Handles MyBase.Load
  'Place Super Bowl Winners into the array
  teamName(0) = "Packers"
  teamName(1) = "Packers"
  teamName(2) = "Jets"
  teamName(3) = "Chiefs"
End Sub
```

Initializing Arrays when they are being created

- Arrays may be initialized when they are created:

```
Dim arrayName() As varType = {value0, _
        value1, value2, ..., valueN}
```

- declares an array having upper bound *N* and assigns *value0* to *arrayName*(0), *value1* to *arrayName*(1), ..., and *valueN* to *arrayName*(*N*).