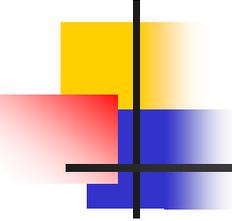
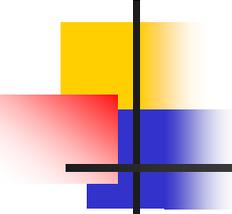


Intro to VBScript



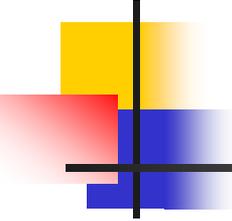
What is VBScript?

An interpreted programming language primarily used to write relatively short programs that access operating system and application objects. Such languages are referred to as "scripting" languages. The VBScript commands are a subset of the more complete, compiled Visual Basic language.



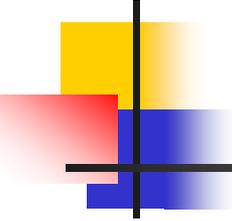
Why is VBScript used in Itec 110?

- VBScript is an easy to learn language and thus is a good candidate for a brief introduction to programming.
- VBScript, VB, and VBA (Visual Basic for Applications) are collectively used more than any other language in the world so exposure to these languages is important.
- VBScript can be used to automate tasks in web pages, MS Excel spreadsheets, MS Access Databases, and other MS applications.
- VBScript can be used to automate administrative tasks such as moving or copying files, creating user accounts, etc.



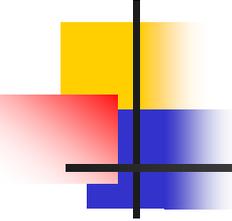
What if I can already program?

- Only about five labs will be used on the intro to programming.
- Take advantage of the opportunity to learn a new language if VBScript is not a language you have used before.
- Take advantage of the opportunity to practice your skills and perhaps pick up some missing pieces in your knowledge of the language.
- If you finish a lab early, challenge yourself by learning features of the language that are new to you and not required in the lab.
- We will try to include a "challenge" task in each VBScript lab.



Creating/Editing a VBScript file

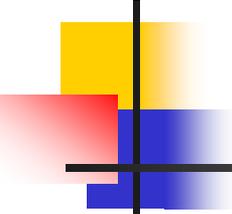
- VBScript programs are ASCII text files and can be edited using any text editor such as notepad.exe or pfe.exe.
- Open your ITEC 110 student folder and create a folder named lab2 within it.
- As was discussed in lab 1 please verify that the tools->folder options->view tab->hide file extensions for known file types is NOT checked.
- Open the lab2 folder. Right click anywhere in the folder and choose "new text file" from the resulting drop down menu. Rename the new text file to HelloWorld.vbs .
- Right click on the new file, choose EDIT, and enter the VBScript program on the next slide.



HelloWorld.vbs

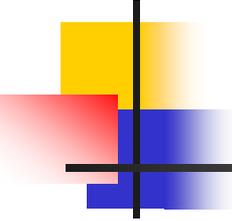
```
' *****  
' HelloWorld.vbs  
' *****  
  
option explicit  
dim strGreeting  
strGreeting = "Hello World"  
Wscript.echo(strGreeting)
```

After you have completed entering the above program close notepad and execute the program by double clicking on it.



Program Variables

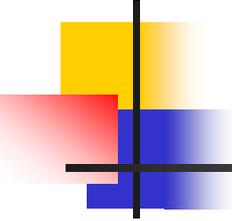
- Variables are containers used to store values. For example `strGreeting` in the program you just wrote is a variable that contains the value "Hello World"
- The most common types of values are integers, floating point numbers, strings, and, Boolean values. Examples of variables being assigned values:
 - Integer `intBattingAverage = 876`
 - Floating Point (single) `sngWeight = 185.3`
 - Floating Point (double) `dblApproxPi = 3.1415926535897932`
 - String `strLastName = "Smith"`
 - Boolean `blnTaxExempt = TRUE`
- Variable names should help explain the significance of the value being stored.
- In the examples above the Leszynski Naming Convention (LNC) is being used. LNC uses a prefix on the variable name to identify the type of value being stored. LNC is not a requirement of VBScript, VB, or VBA but is used by most VB programmers.



Variable Declaration

In most languages variables must be declared before they can be used. VBScript does not require variable declaration by default. It is a good idea to override this default or else debugging your programs will be much more difficult. The program below, which calculates the total price due for the sale of 10 widgets, can be found in the itec110 folder. Copy the program to your lab1 folder and execute it. Does the result look right to you?

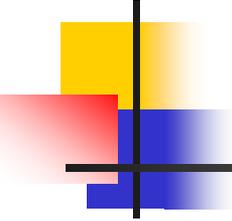
```
' *****  
' NoOpExplicit.vbs  
' *****  
  
Price = 7.99  
Quantity = 10  
Total = Price * Qauntity  
Wscript.echo("The Total is: " & Total)
```



Variables Declaration (2)

Here is the same program with variable declaration required and implemented. Only two lines have been added. On the first line the "option explicit" statement turns on required variable declaration. The second line declares or "dimensions" the variables needed for this program using the DIM statement. Add these first two lines to your widget program and execute it. What is different about the result?

```
Option Explicit
Dim Price, Quantity, Total
Price = 7.99
Quantity = 10
Total = Price * Quantity
Wscript.echo("The Total is: " & Total)
```



Types of Programming Errors

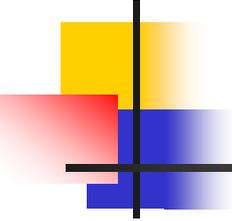
- **Logic Errors:**

The program runs but produces incorrect results.

- **Syntax Errors:**

The program does NOT run but produces an error message indicating that the rules for statement construction (syntax) have been violated.

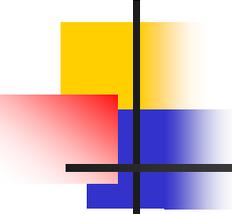
Note: Syntax errors are much easier to detect, locate, and correct than logic errors.



Variables Declaration (3)

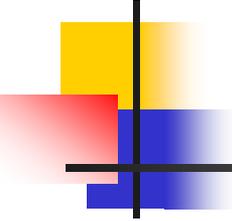
- Variables in VBScript can be declared anywhere in a program as long as it is before that variable is referenced.
- Variables can be declared on one line or many lines.
- Unlike most languages, variables declared in VBScript are "variant" and do not assume a type (string, long, double, integer, boolean) until the first time they are assigned a value.
- Variables in VBScript can be declared using:
 - dim or dimension
 - public
 - private
 - const or constant

Public and Private are keywords which not only declare a variable but help determine its scope. Variable scope determines which parts of a program can "see" and work with a variable. You will learn about variable scope in itec 120. The Const keyword makes a variable a constant which prevents its value from changing after initial assignment (example: `const pi = 3.1415926535897932`)



Assignment Statements

- Assignment is the act of storing a value in a variable. In VBScript the = sign is used for assignment. Examples:
 - `TotalPrice = Quantity x Price`
 - `HoursPerWeek = 7 x 24`
 - `Profit = TotalRevenue - TotalExpenses`
 - `LastName = "Smith"`
- Some languages use := for assignment to differentiate between an assignment statement and an equality. Notice that the VBScript assignment statements below are valid but do not represent equalities.
 - `'increase Counter by 1`
`Counter = Counter + 1`
 - `'increase a customer accounts receivable balance`
`'by the amount of a new sale`
`CustomerBalance = CustomerBalance + NewSaleAmount`

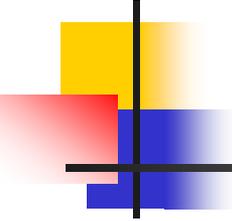


Assignment.vbs

```
' *****  
' Assignment.vbs  
' *****  
  
option explicit  
dim X  
X = 3 * 5  
X = 2 * 10  
X = X + 50  
Wscript.echo (X)
```

What is the final value of the variable X?

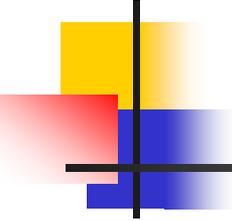
Write and execute this program to find out if you are correct.



Math Operations

The following math operations are available in VBScript:

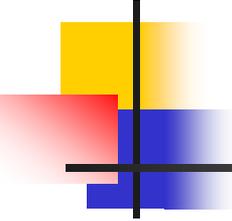
<code>^</code>	exponentiation	<code>xcubed = x^3</code>	9
<code>+</code>	addition	<code>x = 3 + 4</code>	7
<code>-</code>	subtraction	<code>x = 3 - 1</code>	2
<code>*</code>	Multiplication	<code>x = 5 * 30</code>	150
<code>/</code>	division	<code>x = 18 / 3</code>	6
<code>\</code>	integer division	<code>x = 13 \ 5</code>	2
<code>Mod</code>	Modulo	<code>x = 13 mod 5</code>	3



Order of Operations

- Please
- Excuse
- My
- Dear
- Aunt
- Sally

- Parenthesis
- Exponents
- Multiplication
- Division
- Addition
- Subtraction

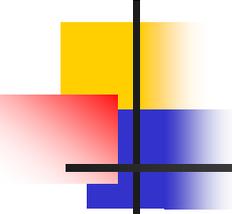


OpOrder.vbs

```
' *****  
' OpOrder.vbs  
' *****  
  
option explicit  
dim X, Y  
X = 3 * 5 + 3 * 5 ^ 2  
Y = (2 * (5 + 3) * 5) ^ 2  
Wscript.echo("The value of X is: " & X)  
Wscript.echo("The value of Y is: " & Y)
```

What is the final value of the variables X and Y?

Write and execute this program to find out if you are correct.

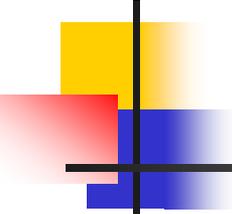


String Concatenation

- Strings in VBScript can be concatenated (connected) using either the + or & operators. Most VBScript programmers use & to avoid confusion with the addition operation:

```
' *****  
' Fullname.vbs  
' *****  
  
Option Explicit  
Dim LastName, FirstName, FullName  
LastName = "Smith"  
FirstName = "Fred"  
FullName = FirstName & " " & LastName  
Wscript.echo(FullName)
```

*What is the purpose of the two quotes in the middle of the concatenation?
Write and execute this program to verify your suspicion.*

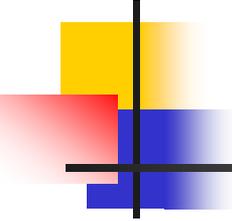


Getting User Input at Runtime

- Programs are much more useful if input can be provided at run time. An easy way to get some input in vbscript is with the InputBox object:

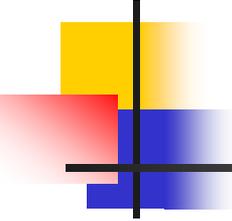
```
' *****  
' FullName2.vbs  
' *****  
  
Option Explicit  
Dim LastName, FirstName, FullName  
LastName = InputBox("Please enter last name: ")  
FirstName = InputBox("Please enter first name: ")  
FullName = FirstName & " " & LastName  
Wscript.echo("The Full Name is: " & FullName)
```

Write and execute this program to see how the InputBox object works.



Comment Lines

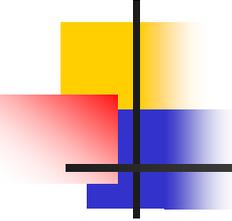
- Comments can be added anywhere in a program using a single quote.
- Comments help document your code and will become more important as your programs get larger and more complex.
- Comment lines are ignored by the interpreter.
- You will typically place comments at the top of your program identifying the program name, creation date, and author. This is sometimes called a "tombstone" or "flower box"



Line Breaks

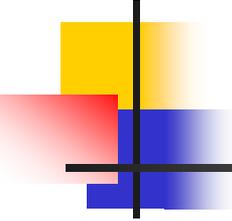
- VBScript uses the hidden CRLF to know where the end of a program statement is. If you have to write a very long program statement you will need to use the underscore _ to act as a continuation. For example:

```
' *****  
' LongString.vbs  
' *****  
  
Option Explicit  
Dim strMessage  
  
strMessage = "This is a string concatenation" &  
"that is too long to fit on a single line of code"  
Wscript.echo(strMessage)
```



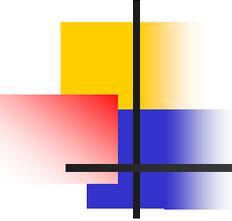
In-lab requirement

- Create a folder named LAB2 in your itec110 student folder.
- Create the two VBScript programs described in the next two slides and make sure they reside in your lab2 folder.
- When writing the programs make sure to use thoughtful variable names
- Include your name, the program name, and the date at the top of each program in a similar manner to the "tombstone" used to comment program names in this presentation.



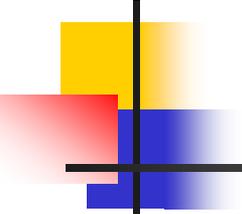
In-lab program 1: KPHtoMPH.vbs

- Create a program which asks the user to enter a speed in kilometers per hour.
- Compute and output the equivalent speed expressed in miles per hour.
- One KPH equals .6MPH



In-lab program 2: Sphere.vbs

- Create a program which asks the user to enter the diameter of a sphere.
- Compute and output the volume of that sphere.
- The formula for the volume of a sphere is $\frac{4}{3}\pi r^3$
- To test your results: a sphere which is 2 units in diameter (regardless of whether those units are feet, inches, cm, etc) is approximately 4.18879 cubic units.
- For this assignment don't worry about how many decimal places are displayed.
- For a challenge also prompt the user to input the units and include the cubed units in the output. Example output: **The volume is 4.18879 cubic feet.**



Homework Assignment

Write a program named loan.vbs that accepts P , APR, n and Y as inputs and then outputs PMT and TI.

$$\text{PMT} = \frac{P \cdot \left(\frac{\text{APR}}{n} \right)}{1 - \left(1 + \frac{\text{APR}}{n} \right)^{-nY}} \quad \text{TI} = (n \cdot Y \cdot \text{PMT}) - P$$

PMT = regular payment

P = starting loan principal (amount borrowed)

APR = annual percentage rate (as a decimal)

n = number of payment periods per year

Y = loan term in years

TI = total interest

Place your loan.vbs program in your lab2 folder no later than the beginning of your next lab period.