CSCI-GA.3033.003 Scripting Languages

9/25/2013

Prelim 1 Review

Outline

- Associativity and Precedence
- Typing
- Properties
- Callbacks

•Concepts

Operator Characterization

•Arity:
•1 = unary
•2 = binary

•Associativity:
•L = left
•R = right

•Precedence:
•from high to low

(…)			all; Indexing
^	2	L	
+, -	1		
*, /	2	L	Multiplicative
\	2	L	Integer division
Mod	2	L	Modulu
+, -	2	L	Additive
&	2	L	String
<<, >>	2	L	Bit shi
=, <>, <, <=, >, >=, Is	2	L	Compari on
Not	1		Negation
And, Or, Xor, Eqv, Imp	2	L	Logic (not all same precedence)
[Set] … = …	2		Assignment statement

•**Concepts**

Arity, Precedence, Associativity

Arity	Number of operands	`-2` `2 - 2`	unary binary

Precedence	Binding strength	`2+2*2`
		~~`(2+2)*2`~~
		`2+(2*2)`
			`*` has higher precedence than `+`

Associativity	Grouping direction	`2/2/2`
		`(2/2)/2`
		~~`2/(2/2)`~~
			`/` is left-associative

- Precedence and associativity in programming usually follows the conventions from math.

Outline

- Associativity and Precedence
- Typing
- Properties
- Callbacks

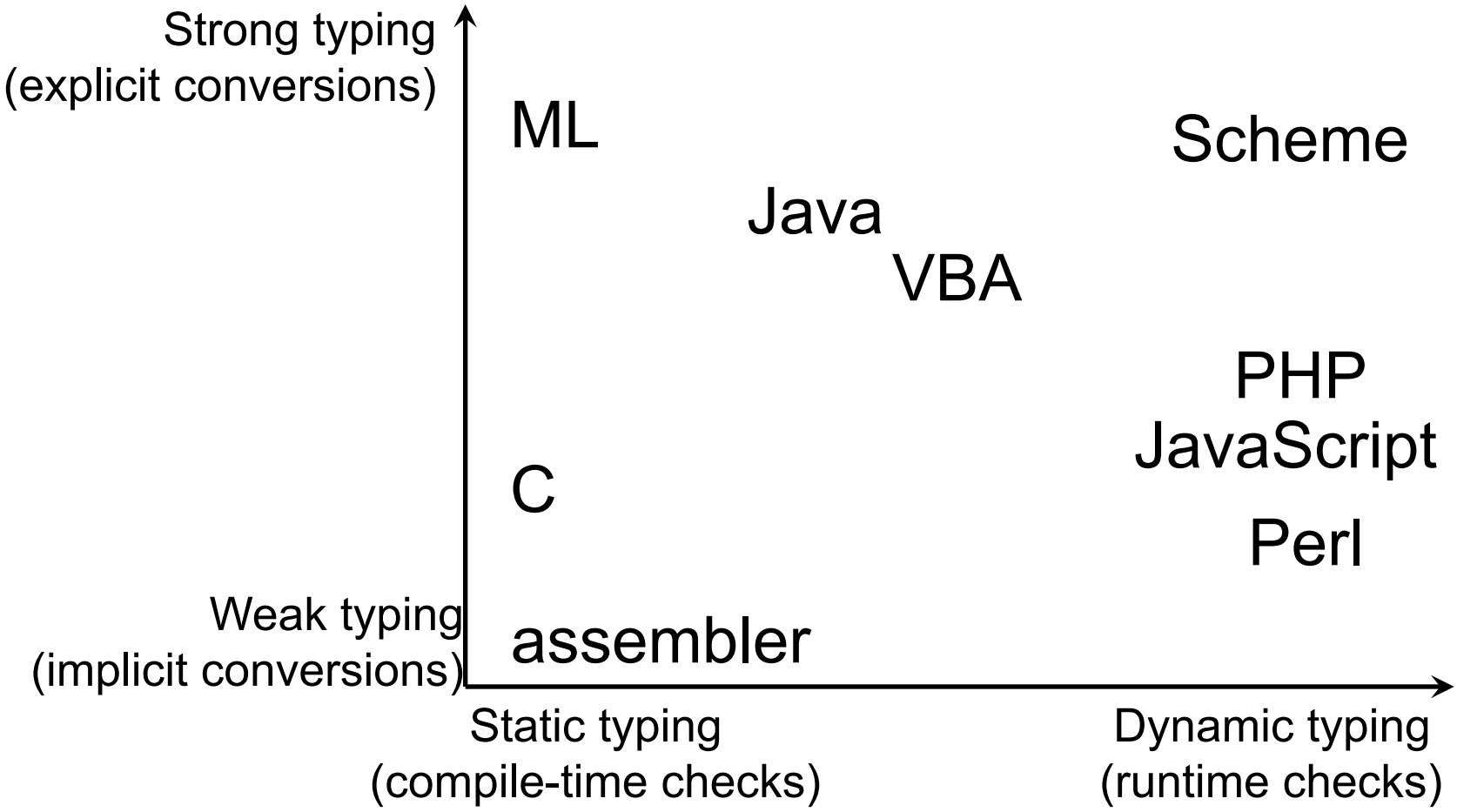
Typing

- **Strong typing** = no implicit type conversion
- **Weak typing** = implicit type conversion
- **Static typing** = check for type errors at *compile* time
- **Dynamic typing** = check for type errors at *run time*
- **Gradual typing** = checks for some errors at compiler time, some at run time. Directed by which parts of the program have explicit types. (Ex. option explicit in VBA)

Typing

- **Explicit typing** = declare the type in your code
  - Java
- **Implicit typing** = compiler infers the type
  - ML language,
  - VBA type declaration characters $ means string

Weak/Strong, Static/Dynamic Typing



Outline

- Associativity and Precedence
- Typing
- Properties
- Callbacks

Properties

- Read and written like fields (dot syntax)
- Accesses are translated to set/get methods
- Have easy-to-read syntax, but can implement complex functionality
- Can be indexed. Seems like an array, but associates a behavior with each read/write

Properties

```
Public Function GetLength() As Double
  GetLength = Sqr(X ^ 2 + Y ^ 2)
End Function
Public Sub SetLength(NewLen As Double)
  OldLen = GetLength
  If OldLen = 0 Then
    X = NewLen
    Y = 0
  Else
    X = X * NewLen / OldLen
    Y = Y * NewLen / OldLen
  End If
End Sub
```

Properties vs. Fields

- Both: dot notation look&feel
  - Writable: **`a1.color = "red`**”
  - Readable: **`Debug.print a1.color`**
- Properties only: active (associated behavior)
  - E.g., update graphical representation
- Properties only: may be indexed, like arrays
  - **`cake.ingredient("topping") = a1`**
- Other languages with properties:
  - E.g., PHP, Delphi, C#

Common Uses of Properties

Simple (field-like)

- Visual update
- Invariant checking
  - Filter illegal values
  - Read-only
  - Copy on write
- Logging

Indexed (array-like)

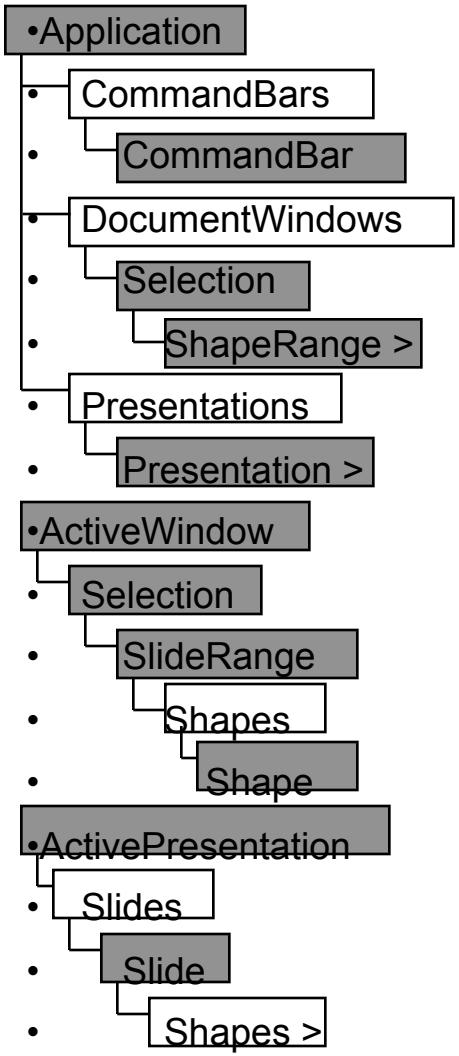
- Collections
  - Resizable array
  - Hash map
- Persistence
  - File
  - Database
  - Cookie

Collections

```
•Dim col As Slides
•Set col = ActivePresentation.Slides
•Dim i As Integer
•Debug.Print "for-loop, indexed property access"
•For i = 1 To col.Count
•   Debug.Print col.Item(i).Name
•Next i
•Debug.Print "for-loop, default property access"
•For i = 1 To col.Count
•   Debug.Print col(i).Name
•Next i
•Dim s As Slide
•Debug.Print "for-each loop"
•For Each s In col
•   Debug.Print s.Name
•Next s
```

Reference

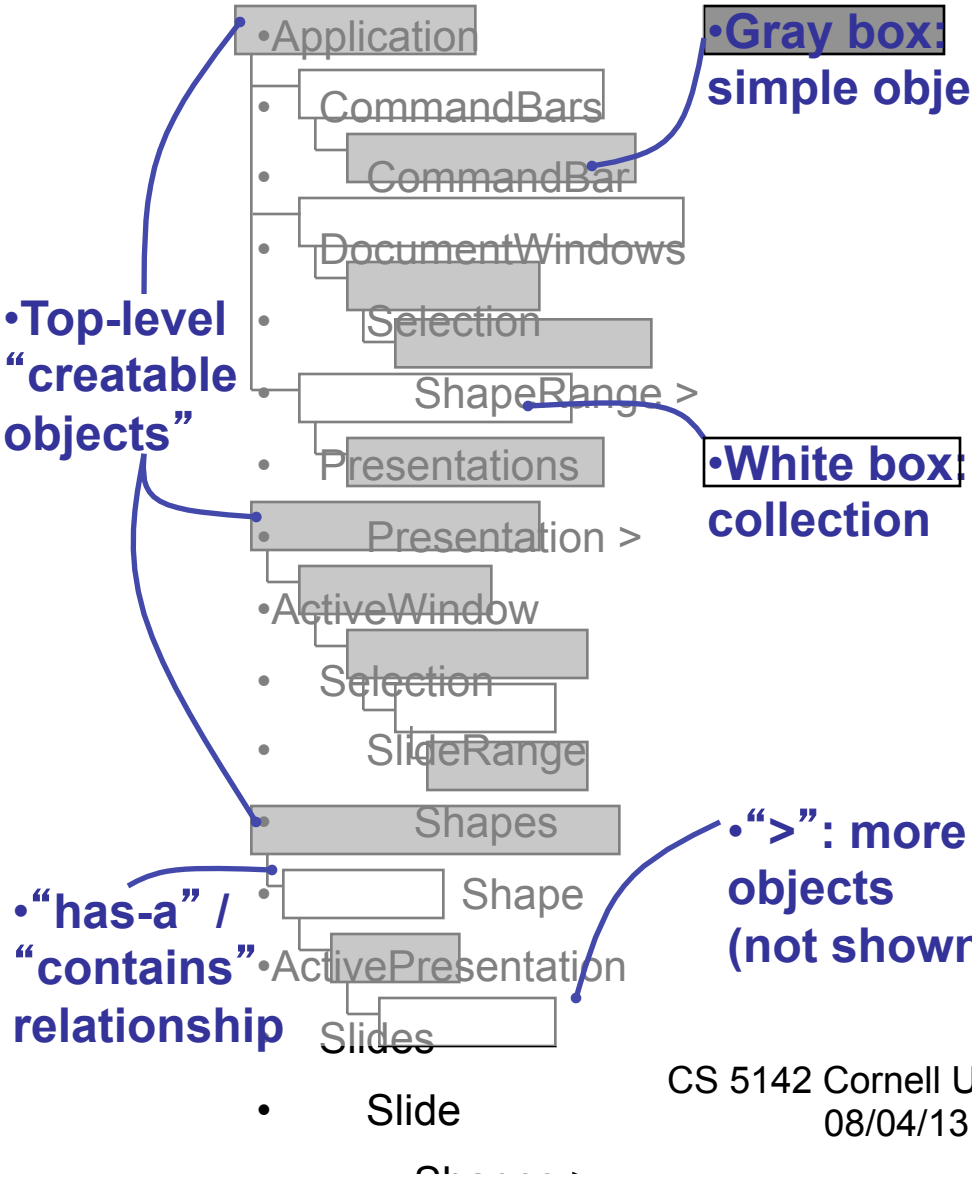
Powerpoint Object Model



- The complete object model is much larger
- See Visual Basic help in editor
- Also in MSDN library:
  - →Office development
  - →Microsoft Office 2003
  - →Office 2003
  - →VBA reference
  - →Powerpoint help
  - →Object model

•Concepts

Object Model



- Object-oriented API for embedded scripts
- Other examples:
  - Object models for other Microsoft apps
  - DOM = document object model for XML

Object Model Usage

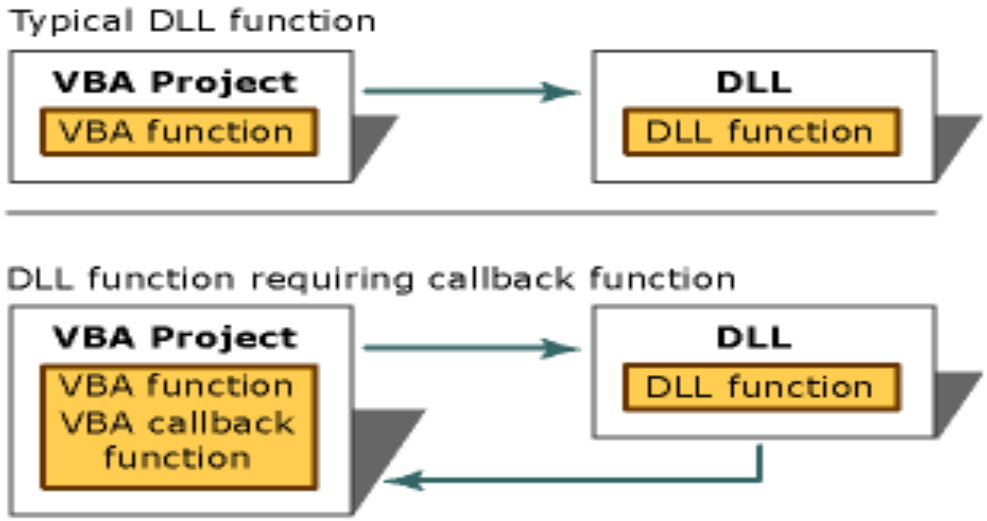
```
Dim S As PowerPoint.Slide
Set S = ActivePresentation.Slides( _
   ActivePresentation.Slides.Count)
```

Outline

- Associativity and Precedence
- Typing
- Properties
- Callbacks

Callbacks

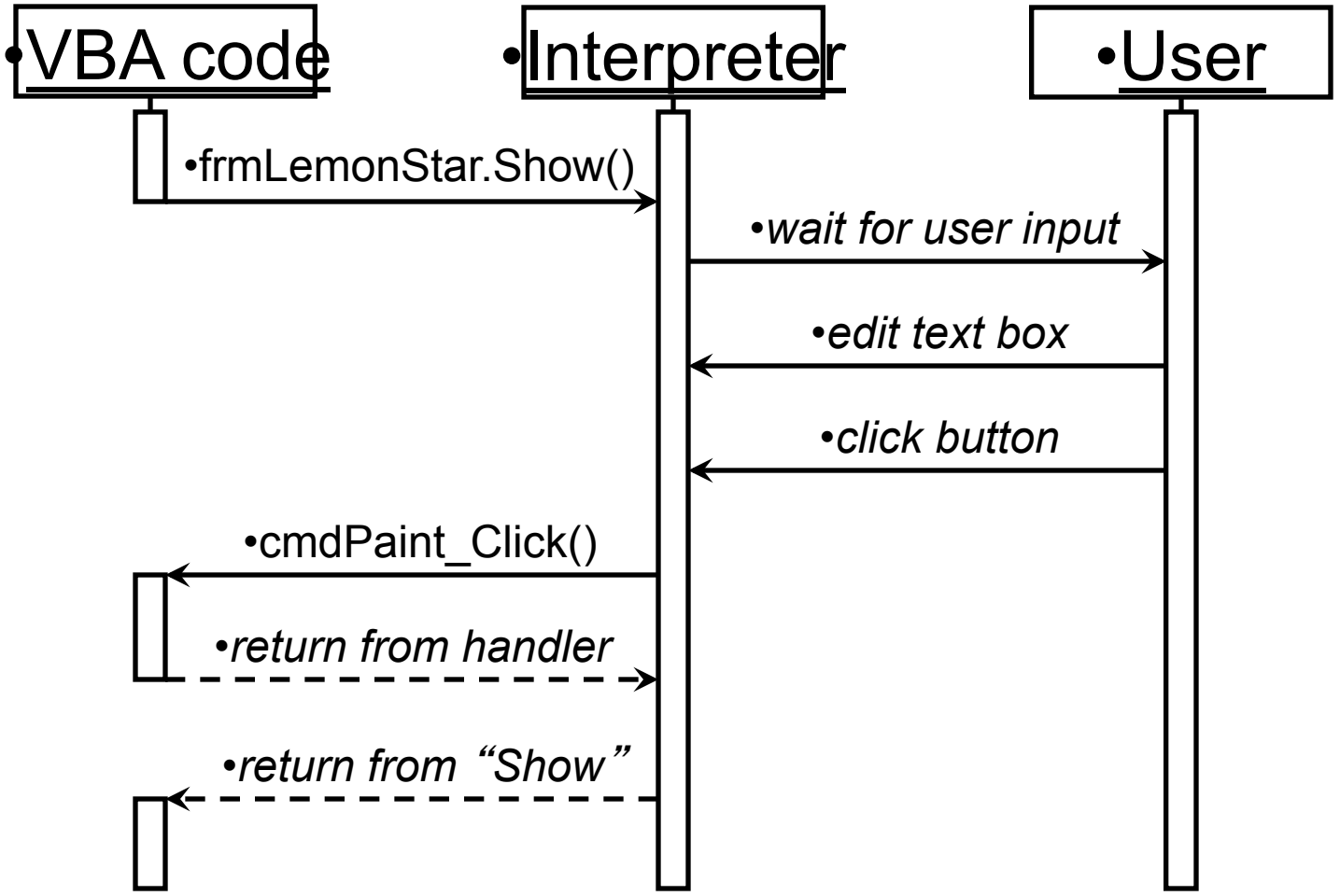
- A *callback* is a function or block of code that is passed to some other code as a parameter.
- It is expected that the callee will “call back” to the called at the appropriate time



Callback Mechanisms

VBA form	Subroutine in form with mangled name
VBA class	**`WithEvent`** / **`RaiseEvent`** statements
Java	Pass object on which to call method
Perl, Python, JavaScript	Pass anonymous function (lambda)
C, C++	Pass function pointer
C++	Pass object on which to call “()” operator
SmallTalk	Pass code block
PHP	Pass name of function as string

Call-backs


•VBA code
•Interpreter
•User
•frmLemonStar.Show()
•wait for user input
•edit text box
•click button
•cmdPaint_Click()
•return from handler
•return from "Show"

Administrative

Last Slide

- Good luck!
- Bring a pencil or pen
- Closed book. No notes, phones, etc.