CS 5142
Scripting Languages

Introduction

Today’s Outline

- Introduction to Scripting Languages
- Course Mechanics

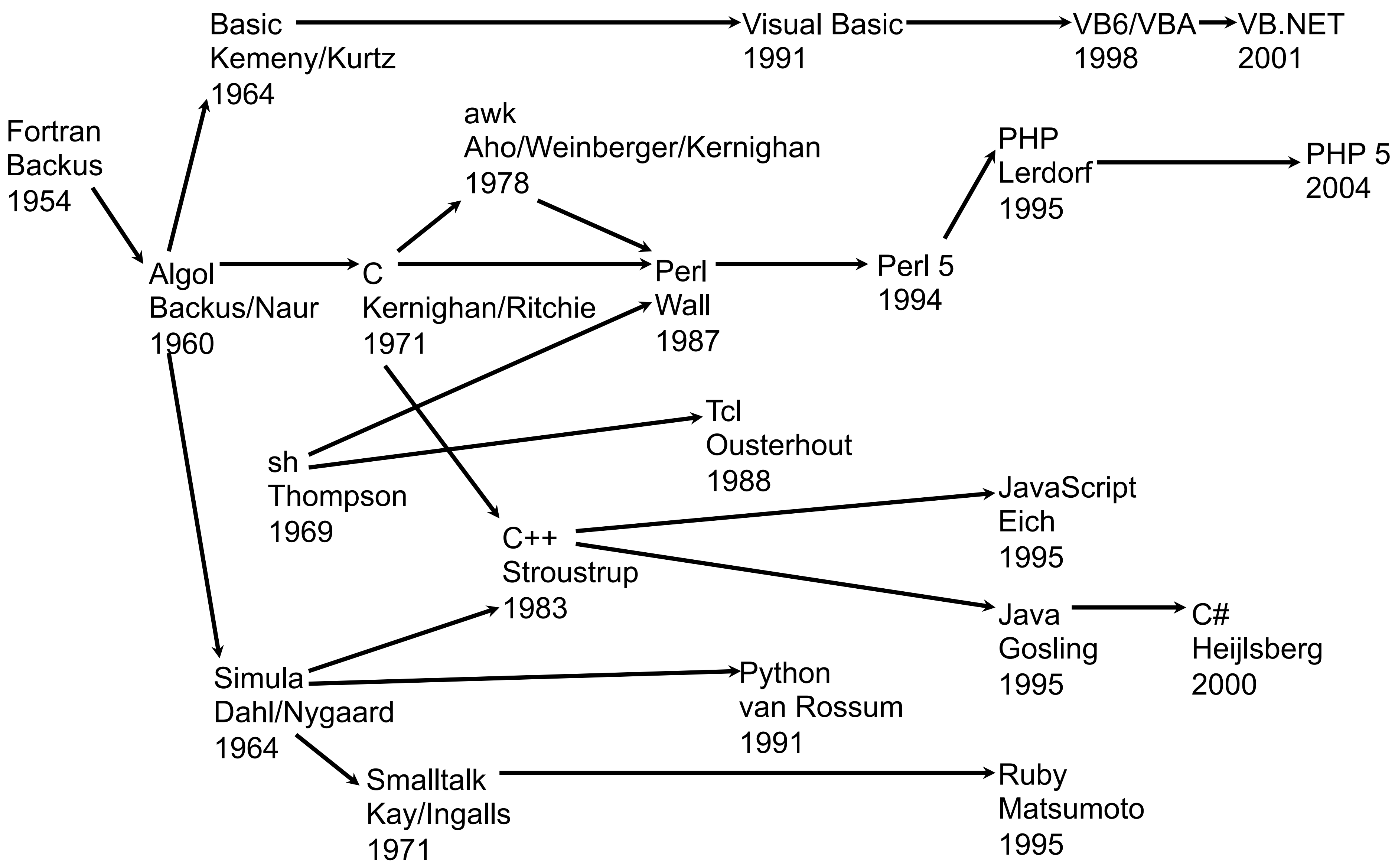
Scripting Languages

Facebook is written in PHP

Amazon makes heavy use of Perl CGI

BBC reports that 73% of all websites use JavaScript

History of Languages



Fortran
Backus
1954

Algol
Backus/Naur
1960

Basic
Kemeny/Kurtz
1964

Visual Basic
1991

VB6/VBA
1998

VB.NET
2001

C
Kernighan/Ritchie
1971

awk
Aho/Weinberger/Kernighan
1978

Perl
Wall
1987

Perl 5
1994

PHP
Lerdorf
1995

PHP 5
2004

sh
Thompson
1969

Tcl
Ousterhout
1988

C++
Stroustrup
1983

JavaScript
Eich
1995

Java
Gosling
1995

C#
Heijlsberg
2000

Simula
Dahl/Nygaard
1964

Python
van Rossum
1991

Smalltalk
Kay/Ingalls
1971

Ruby
Matsumoto
1995

Systems Languages

- Introduced as an alternative to assembly language
- Provide “higher level” instructions
  - 1 line of C code ≈ 3-7 assembly instructions
- Are strongly typed
  - The “type” determines how data can be used

Advantages of Strong Typing

- Can catch errors at compile time
  - Example: using a floating point instead of a pointer
- Can make large code bases more manageable
  - Clarify how things are used
- Compiler can use type information to generate more efficient code
  - Example: can generate integer instructions

Scripting Languages

- Often assume the existence of components which they “glue” together
- Different target programmer
- Tend to be weakly typed
- Usually interpreted (not compiled)
  - Tradeoff performance for expressiveness
- Encourage rapid prototyping and development

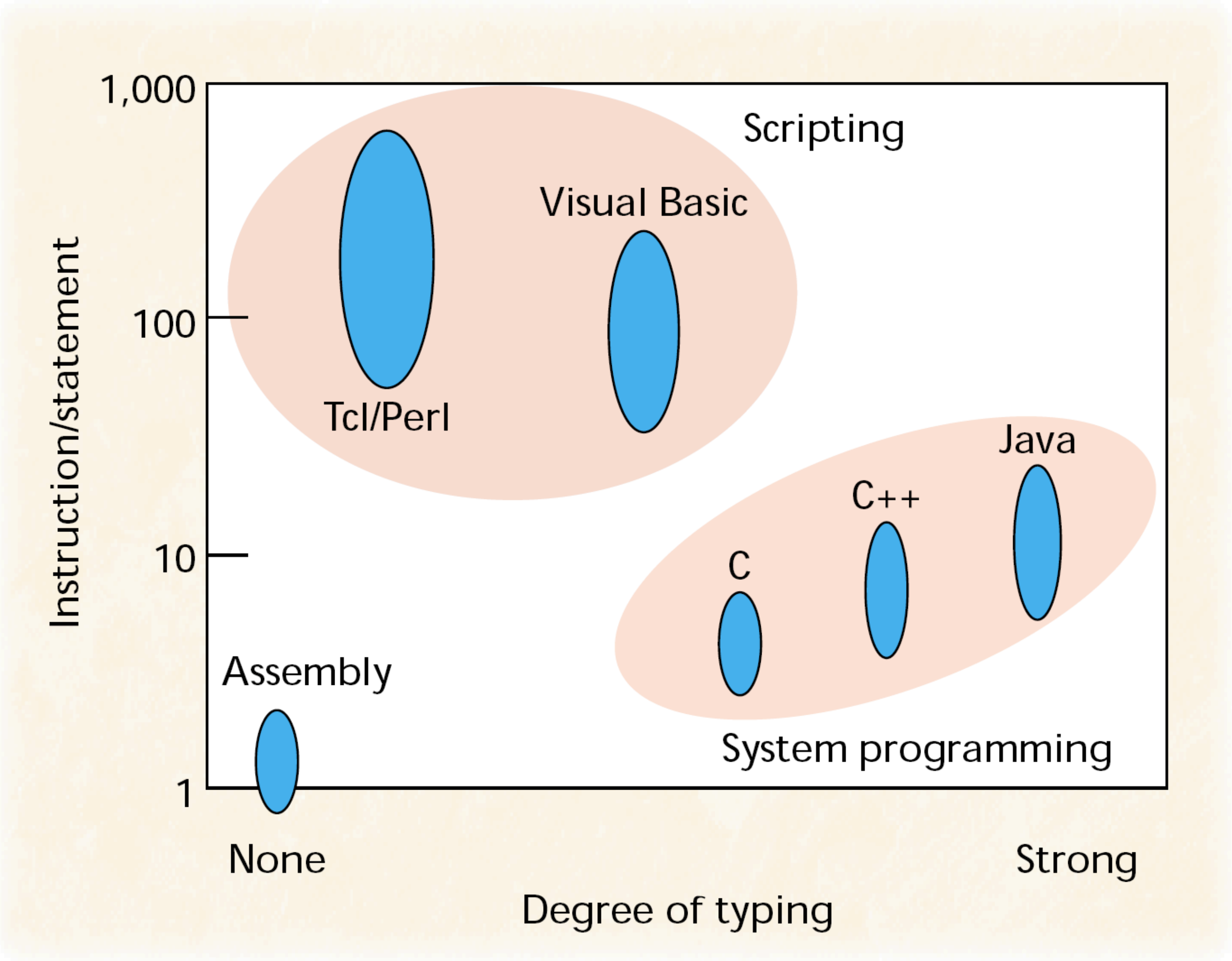
Advantages of Weak Typing

- No a-priori restrictions
- Easier to “hook things together”
  - Example: Unix shell commands, everything is just a string
- Encourages code re-use
  - Don’t need different interfaces for different types
- More succinct code

Features for Rapid Development

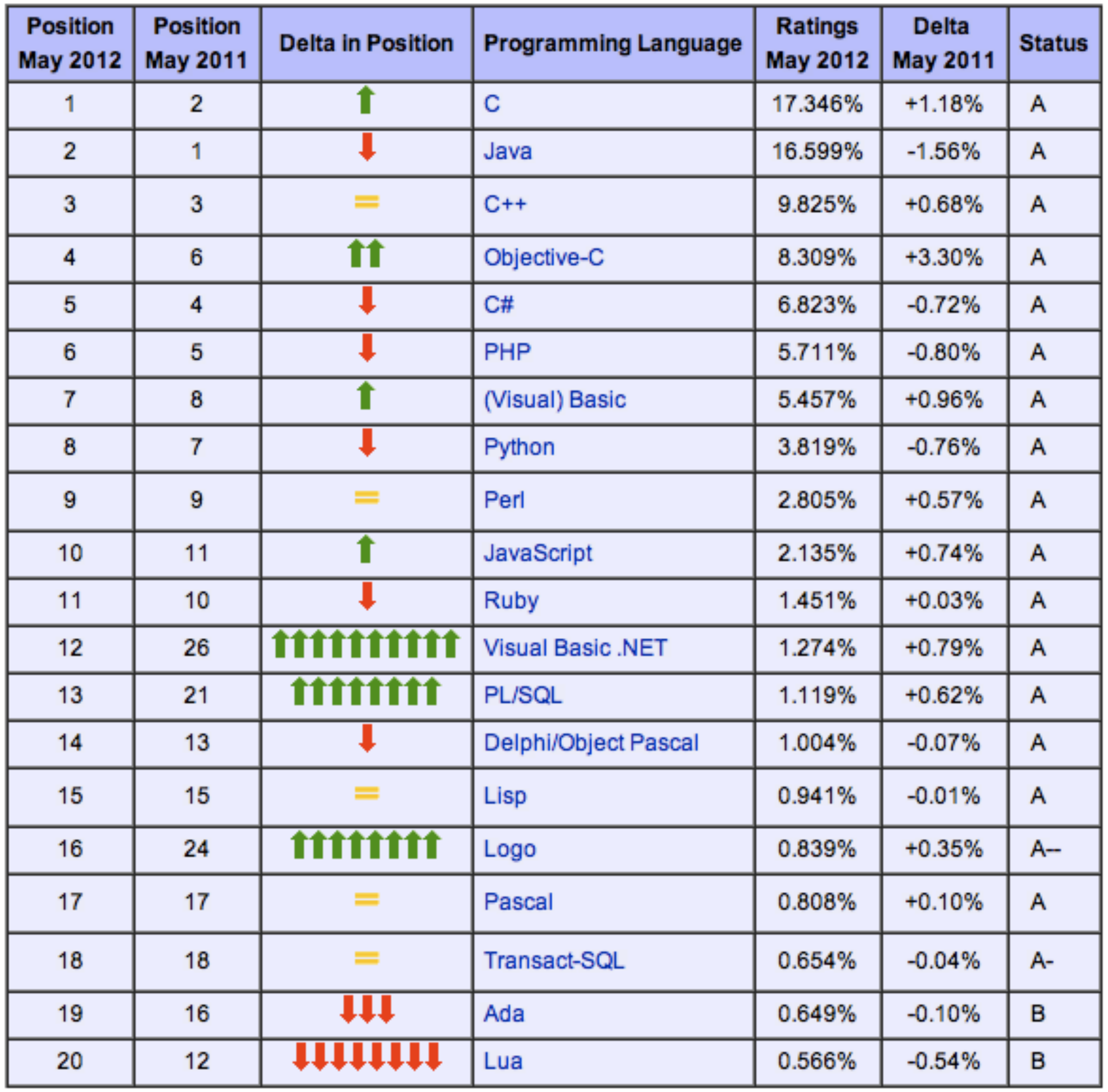
Low boiler-plate	`print "Ice cream!\n"`
Dynamic typing	`$amount = 20 . " grams";`
Interpretation	`eval "print 'egg\n'";`
String manipulation	`$x = "food"; $x =~ s/o/e/g;`
Associative arrays	`$group{pasta} = 'carbs';`
Properties	`document.im1.src="meal.jpg";`
Call-backs	`<input … onClick="stir()">`

Scripting vs. Systems Languages



Source: [Ousterhout'98] http://www.tcl.tk/doc/scripting.html

Language Popularity Index



Position May 2012	Position May 2011	Delta in Position	Programming Language	Ratings May 2012	Delta May 2011	Status
1	2	⬆	C	17.346%	+1.18%	A
2	1	⬇	Java	16.599%	-1.56%	A
3	3	=	C++	9.825%	+0.68%	A
4	6	⬆⬆	Objective-C	8.309%	+3.30%	A
5	4	⬇	C#	6.823%	-0.72%	A
6	5	⬇	PHP	5.711%	-0.80%	A
7	8	⬆	(Visual) Basic	5.457%	+0.96%	A
8	7	⬇	Python	3.819%	-0.76%	A
9	9	=	Perl	2.805%	+0.57%	A
10	11	⬆	JavaScript	2.135%	+0.74%	A
11	10	⬇	Ruby	1.451%	+0.03%	A
12	26	⬆⬆⬆⬆⬆⬆⬆⬆⬆⬆	Visual Basic .NET	1.274%	+0.79%	A
13	21	⬆⬆⬆⬆⬆⬆⬆⬆	PL/SQL	1.119%	+0.62%	A
14	13	⬇	Delphi/Object Pascal	1.004%	-0.07%	A
15	15	=	Lisp	0.941%	-0.01%	A
16	24	⬆⬆⬆⬆⬆⬆⬆⬆	Logo	0.839%	+0.35%	A--
17	17	=	Pascal	0.808%	+0.10%	A
18	18	=	Transact-SQL	0.654%	-0.04%	A-
19	16	⬇⬇⬇	Ada	0.649%	-0.10%	B
20	12	⬇⬇⬇⬇⬇⬇⬇⬇	Lua	0.566%	-0.54%	B

- 5/10 most popular languages are scripting
- This class will cover the top 4 most popular (excluding Python)

Text Processing Example

```
#!/usr/bin/perl -w
%cup2g = ( flour => 110, sugar => 225, butter => 225 );
%volume = ( cup => 1, tbsp => 16, tsp => 48, ml => 236 );
%weight = ( lb => 1, oz => 16, g => 453 );
while (<>) {
  my ($qty, $unit, $ing) = /([0-9.]+) (\w+) (\w+)/;
  if ($cup2g{$ing} && $volume{$unit}) {
    $qty = 1.0 * $qty * $cup2g{$ing} / $volume{$unit};
    $unit = 'g';
  } elsif ($volume{$unit}) {
    $qty = 1.0 * $qty * $volume{ml} / $volume{$unit};
    $unit = 'ml';
  } elsif ($weight{$unit}) {
    $qty = 1.0 * $qty * $weight{g} / $weight{$unit};
    $unit = 'g';
  }
  printf("%d $unit $ing\n", $qty + .5);
}
```

Application Extension Example

```
Option Explicit

Sub LemonStar()
  Dim S As PowerPoint.Slide
  Set S = ActivePresentation.Slides( _
    ActivePresentation.Slides.Count)
  Dim I As Integer
  For I = 0 To 8
    Dim L As PowerPoint.Shape
    Const Dpi As Integer = 72 ' 72 dots per inch
    Set L = S.Shapes.AddLine( _
      BeginX:=Dpi*5, BeginY:=Dpi*3.75+I*Dpi/8, _
      EndX  :=Dpi*6, EndY  :=Dpi*4.75-I*Dpi/8)
    L.Line.ForeColor.RGB = RGB(I * 31, I * 31, 0)
  Next I
End Sub
```

Server Side Scripting Example

```
<?php

$d = sqlite_open("data/sqlite2", 0666, $err);
if ($err) { die($err); }
sqlite_query($d, "select * from T", SQLITE_BOTH, $err);
if ($err) {
  echo "table does not yet exist, creating it …<br>";
  $q = "create table T(I integer, S char(10))";
  sqlite_query($d, $q, SQLITE_BOTH, $err);
  if ($err) { die($err); }
  sqlite_query($d, "insert into T values(0, 'n')");
}
$rows = sqlite_query($d, "select I from T where S='n'");
$row = sqlite_fetch_array($rows, SQLITE_BOTH);
echo "T[S=n][I]==" . $row['I'] . "; reload for ++<br>";
sqlite_query($d, "update T set I = I+1 where S='n'");
echo "delete data/sqlite2 to start over<br>";
?>
```

Client Side Scripting Example

```
<html>
<head><title>Form validation example</title>
<script>
function chk() {
  var v = document.myFm.num.value;
  if (v>=1 && v<=10) return true;
  alert("bad input " + v);
  return false; //abort commit
}
</script>
</head><body>
<form name="myFm" method="post" action="otherpage.htm">
Enter a number: <input size="4" type="text" name="num">
<input type="submit" value="OK" onClick="return chk()">
</form></body>
</html>
```

Today's Outline

- Introduction to Scripting Languages
- Course Mechanics

Course Goals

- Short-term:
  - Survey of the most popular scripting languages
  - Understand general PL concepts in the context of scripting
- Long-term
  - Use languages effectively
  - Quickly learn new languages on your own
  - Design and improve scripting languages

Tentative Schedule

Lecture topic
Introduction
End-user programming (VBA)
Objects, properties, call-backs
Textual data processing (Perl)
Contexts, objects, scripting as glue
Server-side scripting (PHP)
Client-side scripting (JavaScript)
Web applications and databases
Security for web applications
Other languages (Bash, Python, Ruby)

Grading

- 25% homework + 35% prelims + 40% final
- Homework
  - Due each Friday at 6pm
  - >= 1 minute late: 50% points
  - >= 3 hours late: 0%
- Contact me for circumstances beyond your control

Academic Integrity

- Please see: http://cuinfo.cornell.edu/Academic/AIC.html
- You may collaborate on homework, *but*:
  - You must write up and turn in your own answers
  - You must indicate who you collaborated with

Contact

- Instructor: Robert Soulé
  soule@cs.cornell.edu
- Office hours: Fridays after class (or by appointment)
- TA: To be determined.
- http://www.cs.cornell.edu/Courses/cs5142/2013fa/

Recommended Books

- No required textbooks
- If you want more detail, there is a list of recommended books on the course website

Next Time

- End-User programming
- Introduction to VBA
- **You will need access to Microsoft Powerpoint**
  - **Email me if you can’t get it**