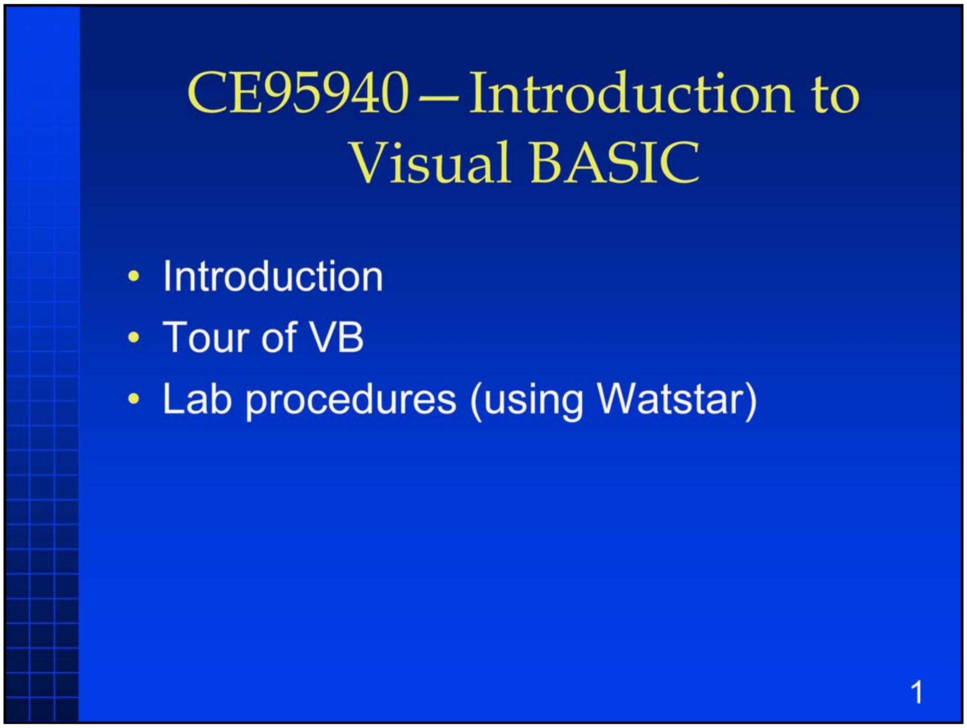


CE95940 — Introduction to Visual BASIC

- Introduction
- Tour of VB
- Lab procedures (using Watstar)

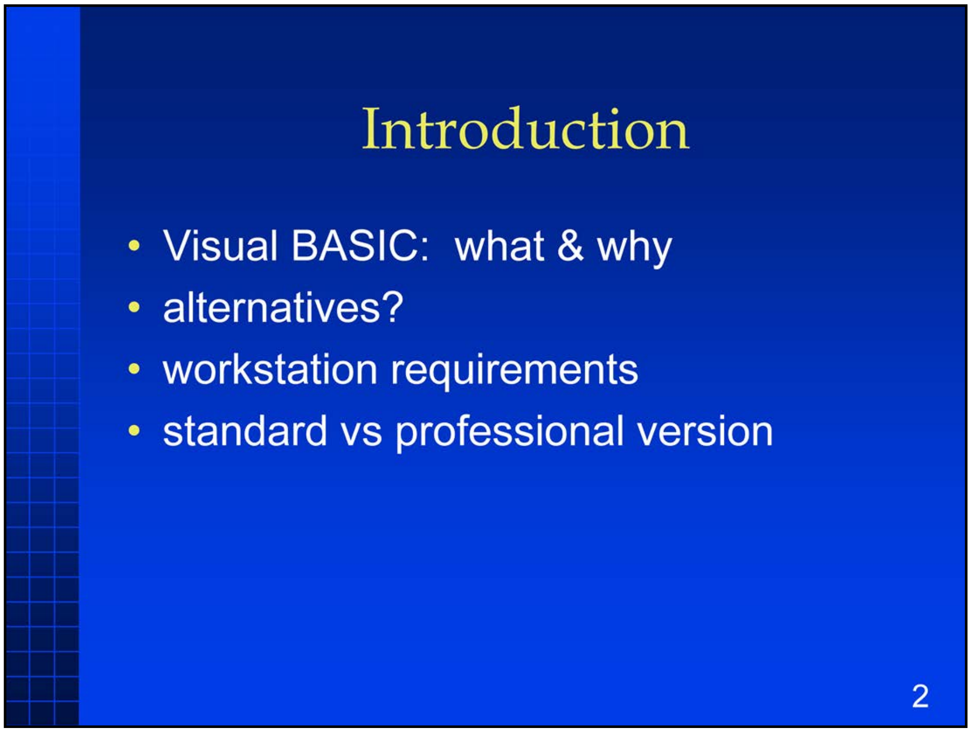
General introduction -- outline structure of classes

Distribute materials -- advise of individuality of handouts

Interactive classes -- questions welcome, discussion encouraged

Background survey: Windows users, programming, already tried VB

Interest survey: DB, other platforms



Introduction

- Visual BASIC: what & why
- alternatives?
- workstation requirements
- standard vs professional version

VB -- development system for Windows applications

Easy to create sophisticated applications -- interactive, seamless development environment

MS estimates 75% of all VB apps used for database

Short learning curve -- hours/days instead of weeks/months (e.g. standard windows API has 3000+ entrypoints)

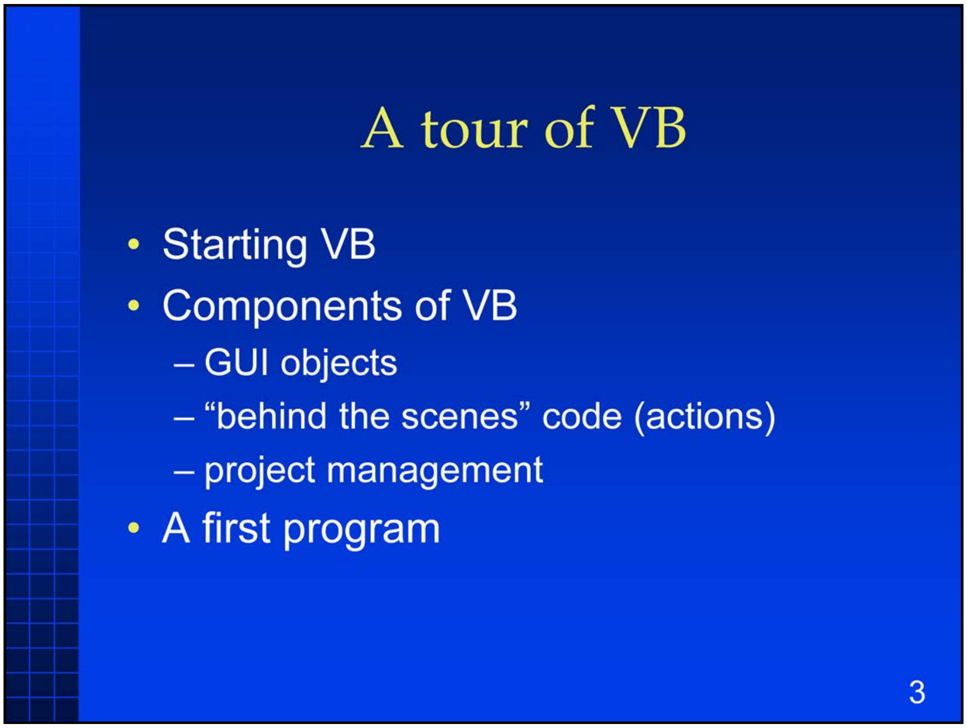
Disadvantage -- slower applications, restricted range of Windows bells & whistles (maybe an advantage!)

But -- extensible (can call Windows API functions, create new function packages

Many alternatives for specific application domains (database, multimedia, cross-platform) -- VB is a good general-purpose tool. Will try to arrange guest speakers for some alternatives

Requirements: Windows 3.1 -- for development, double MS estimates (486, 8Mb); screen resolution (real estate)

Standard vs prof.: prof == standard + DB + bells & whistles



A tour of VB

- Starting VB
- Components of VB
  - – GUI objects
  - – “behind the scenes” code (actions)
  - – project management
- A first program

VB provides a design tool to simplify creating forms/windows and attaching actions

Start VB

Overview of components: VB menubar, toolbox, properties, project (module list), code windows

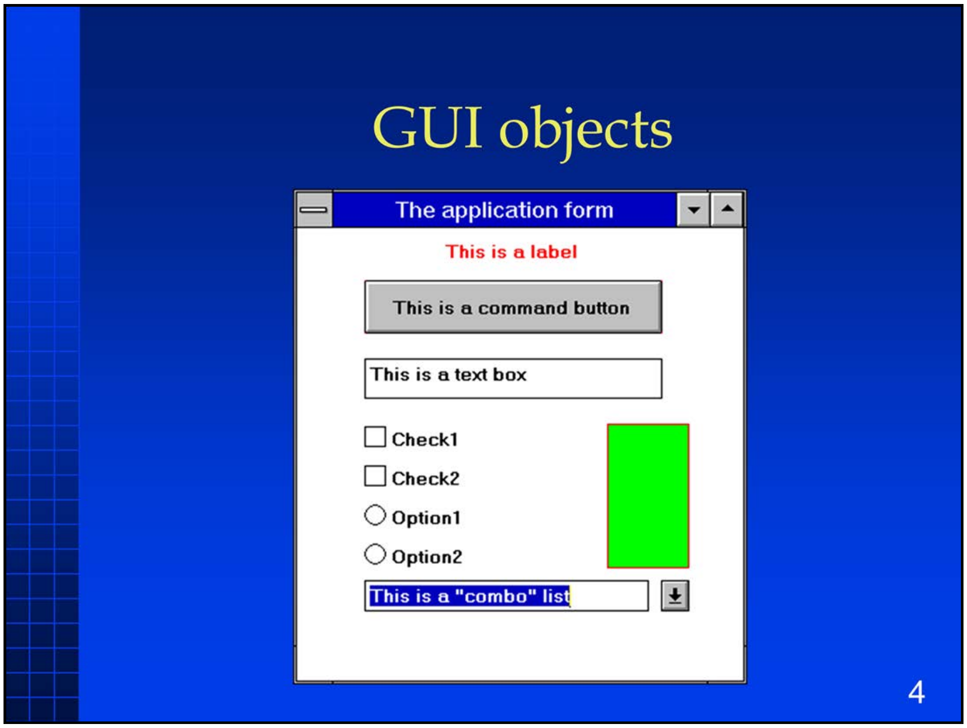
write a “press me” program -- nop; don’t show code

GUI objects: buttons, text, lists, pictures, etc. Forms & windows terminology

VB provides development environment to facilitate creating code to go with the visual interface

Project management: create executables, manage resources. As application grows, manage code modules

Save the program in a new directory



GUI objects

The application form

This is a label

This is a command button

This is a text box

□ Check1

□ Check2

○ Option1

○ Option2

This is a "combo" list

Others include simple lists, sliders, pictures, timers, file-system controls

Properties/attributes include colour, text/caption, set/not set, elements in the list size, fonts, name

Hundreds of different properties, many with restricted ranges

Design tool helps to simplify creation of windows -- object property window

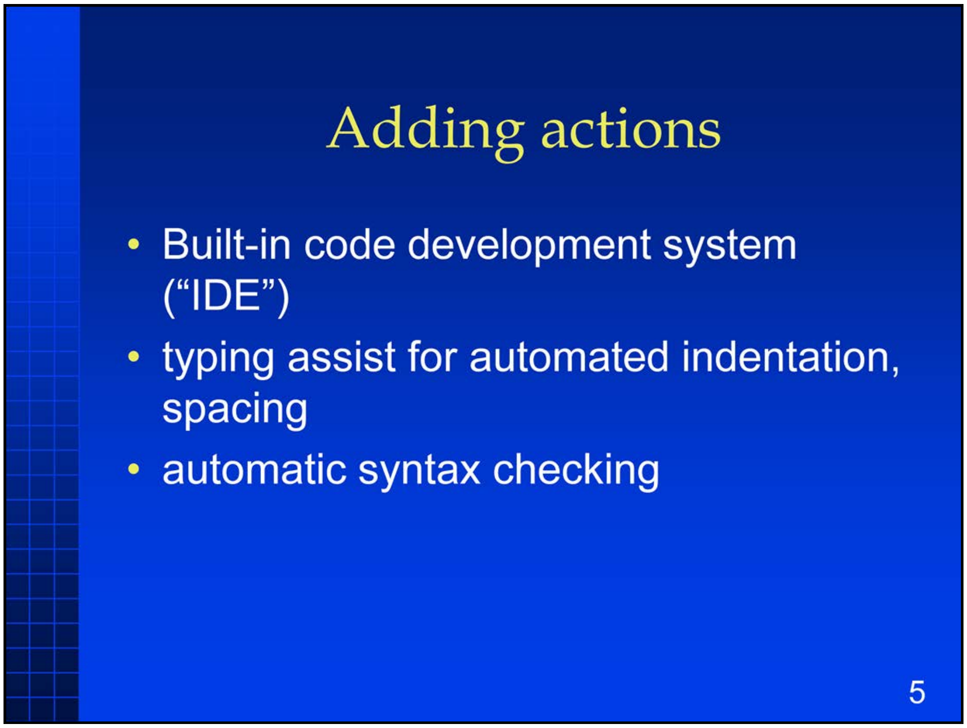
Grid allows easy alignment of objects on form: adjustable size

Objects have events: click, double-click, change (text), selection (lists), other specialized ones -- click is most common. These generally not same as Windows events, simplified

Attach code (actions) to events: display information, read contents of object, disk-file operations

Flip to development, demo 2, and fiddle: move/change objects -- no actions, use properties window

use context help for objects



Adding actions

- Built-in code development system (“IDE”)
- typing assist for automated indentation, spacing
- automatic syntax checking

adding actions requires writing code -- VB provides an IDE to help make coding easier.

colour used for highlighting syntactic elements (keywords, function names, comments) -- configurable to some degree

local/module/global search/change

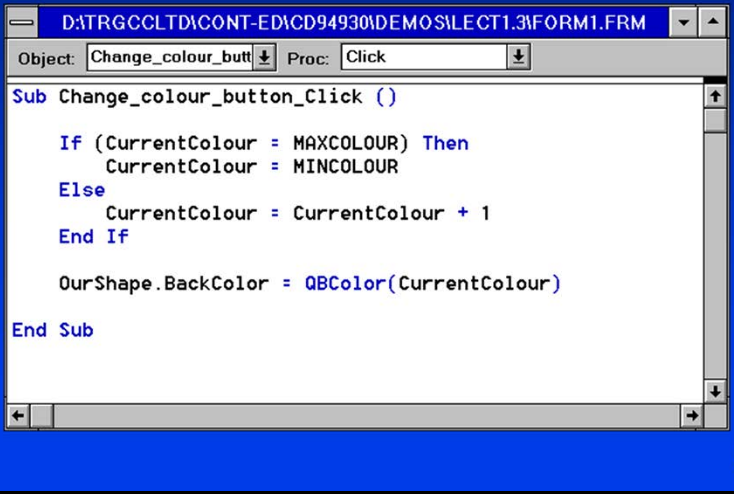
module management, code management

IDE is not mandatory but awkward to work with external

IDE not brilliant but generally usable

Example: press me button changes its caption: click event has procedure associated. Explain how to get to event routines (via object, via code-window) Explain dropdowns in code-window

IDE environment


D:\TRGCCLTD\CONT-ED\CD94930\DEMOS\LECT1.3\FORM1.FRM
Object: Change_colour_butt
Proc: Click

```
Sub Change_colour_button_Click ()

    If (CurrentColour = MAXCOLOUR) Then
        CurrentColour = MINCOLOUR
    Else
        CurrentColour = CurrentColour + 1
    End If

    OurShape.BackColor = QBColor(CurrentColour)

End Sub
```

Slightly more complex example...

load colour demo, run

look at code for colour-change click -- what are CurrentColour, MAXCOLOUR, etc

show declarations and explain.

hand-wave about 16 colour function from quickbasic, want full colour mix of r, g, b

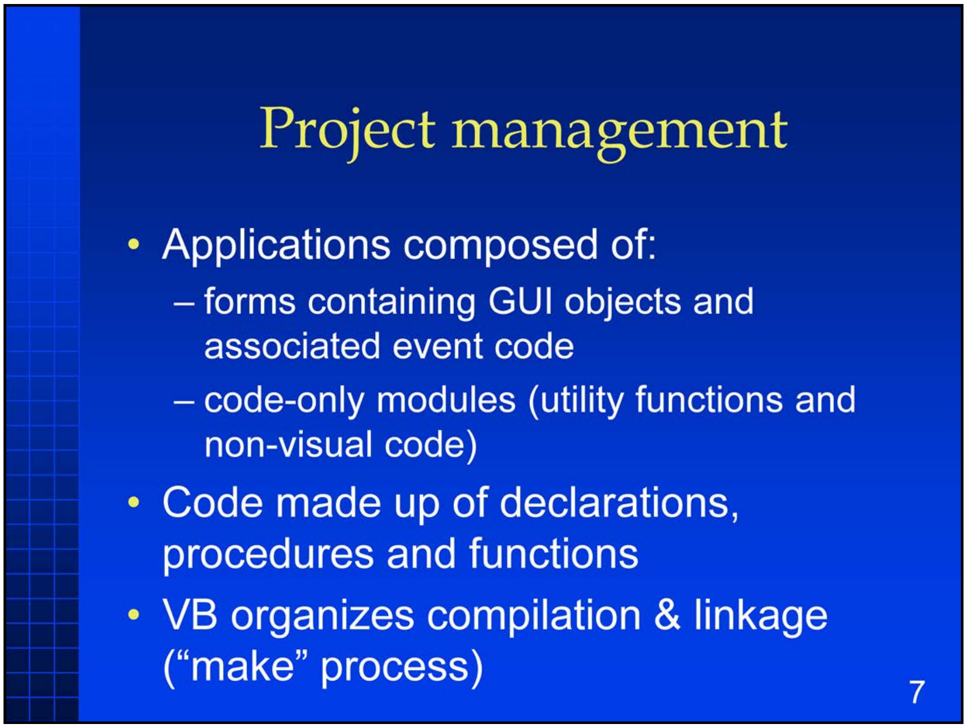
change QBColor to RGBColor, add two zero parms (just varies red)

Observe keyword colour doesn’t change: not correct. Change to “rgb” (case change?)

Run and lean on return -- dark to light shade of red. Doesn’t work!

change max colour to 255

demolnstrate context help for keywords, functions



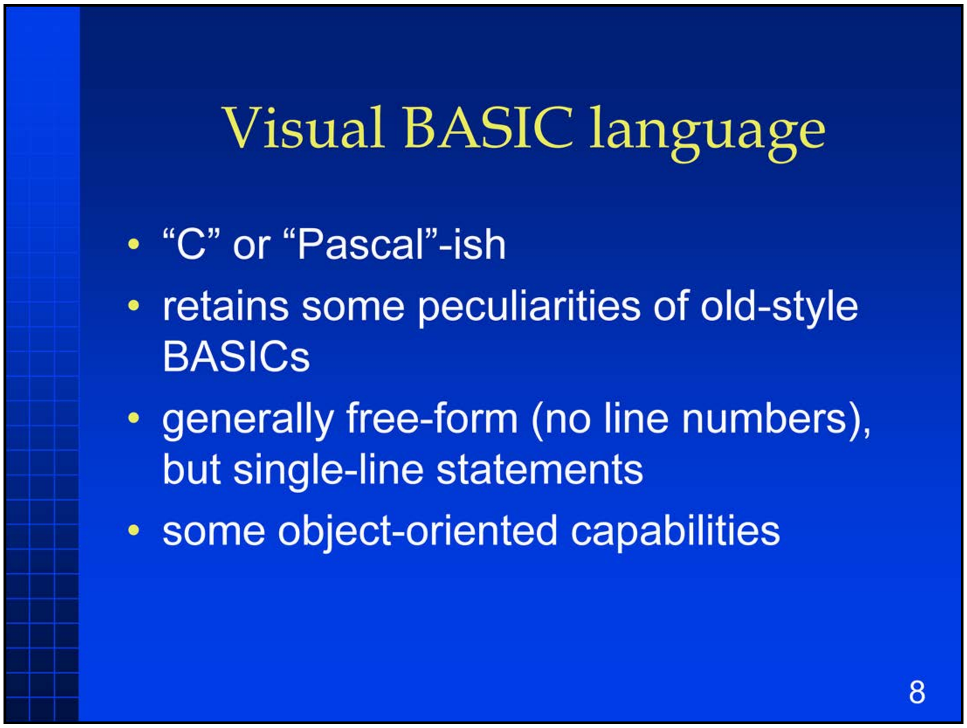
Project management

- Applications composed of:
  - forms containing GUI objects and associated event code
  - code-only modules (utility functions and non-visual code)
- Code made up of declarations, procedures and functions
- VB organizes compilation & linkage (“make” process)

hand-wave

mention file types: .mak for project org; .frm for forms and associated code; .bas for code-only

also: .frx for icons attached to form



Visual BASIC language

- “C” or “Pascal”-ish
- retains some peculiarities of old-style BASICs
- generally free-form (no line numbers), but single-line statements
- some object-oriented capabilities

not the 1960’s or 1970’ flavour of language

resembles Pascal or C -- full set of control structures (if, while, for, select, etc..)

procedures & functions, call by reference & call by value (default is reference)

var declarations optional -- omission strongly discouraged (use “option explicit”)

can use var-name suffices to show types -- discouraged. Prefer to use declarations (all done with “dim”)

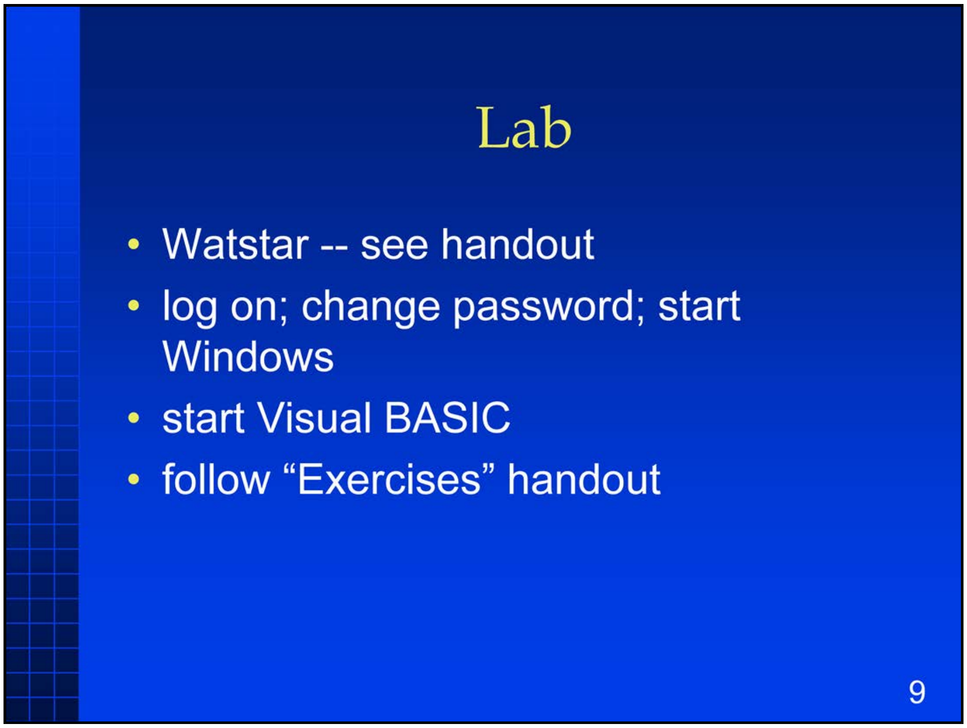
standard datatypes: integer, long, real, double, string, arrays, user-defined (Pascal records or C structs)

constant declarations -- readonly

no dynamic memory or pointers, but dynamic run-time arrays

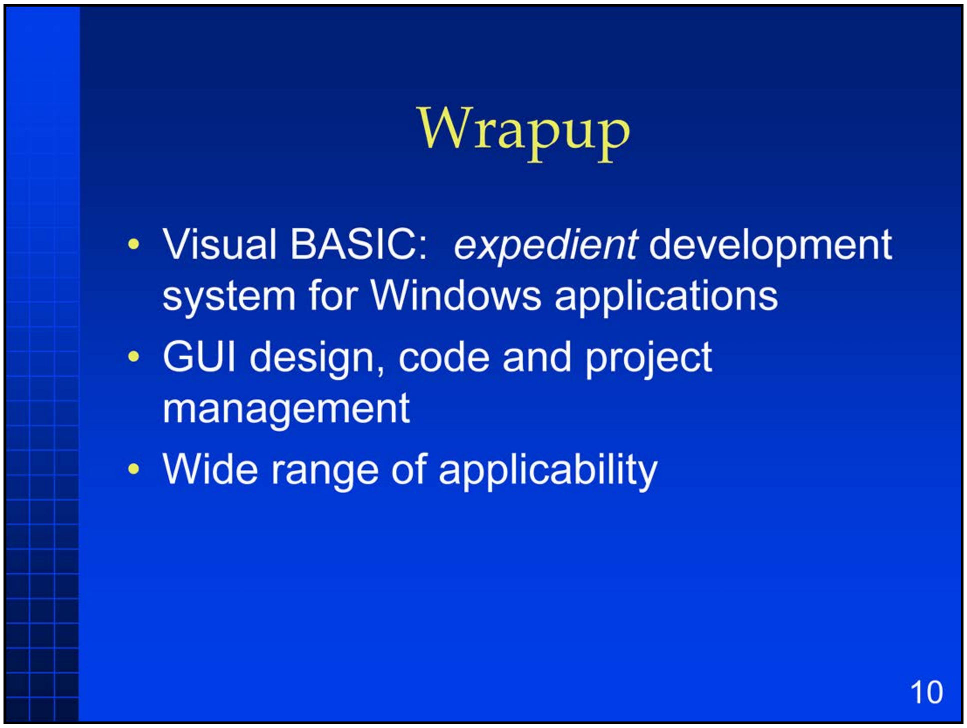
object types and references, object instantiation; no operator overloading or subclassing

online help is very useful -- use hypertext



Lab

- Watstar -- see handout
- log on; change password; start Windows
- start Visual BASIC
- follow “Exercises” handout



Wrapup

- Visual BASIC: *expedient* development system for Windows applications
- GUI design, code and project management
- Wide range of applicability

Not perfect, but resoundingly successful

MS grossly underestimated market acceptance -- 50-page book of add-on controls