COSC 1530

Introduction to Computer Use II: Programming

Winter 2005 (Section M)

Topic A: Introduction to Problem Solving and Visual Basic

Friday, January 13 2006

Bill Kapralos

COSC 1530, Winter 2006, Bill Kapralos

Overview (1):

- **Before We Begin**
  - Some administrative details
  - Some questions to consider
- **Introduction to Programming with VB 6.0**
  - Event handling
  - General programming principles
  - "Live" examples/demo

Before We Begin

Administrative Details (1):

- **Once Again, Lab Exercise 2-3**
  - From your textbook
  - Follow instructions given on the course website
  - Due Monday, January 16 2005 before noon
    - Place in the assignment drop-box located on the 1st floor of the CSE building just by the elevator and CSE undergraduate offices
  - I will drop by the Glade lab after today's lecture
  - If you experience any troubles with the equipment in the Glade lab, go over to CNS in the William Small Centre

---

Some Questions to Consider (1):

- What is a form ?
- What is a control ?
- What is a property ?
- What are the two ways to set/change a property ?
- What must we do in order to handle events ? In other words, why is that if I simply place a control button on the form, run the program and click on the button nothing will happen ?

Programming with MS Visual Basic 6.0 (cont.)

Review - Controls (1):

- **How do we Access/Specify Control Properties within the Code ?**
  - "Standard" convention → Control_Name.Property
    - Control_Name → the name of the control (e.g., "Label1")
    - "." → period
    - Property → the desired property (e.g., "Caption")
  - Example
    - Label1.Caption → refers to the Caption property of the control called "Label1"
    - Can assign it a value → Label1.Caption = "Hello"

Event Handling (1):

- **Control Event Handling**
  - Controls also have a mechanism for handling the many different types of possible user events
    - Example → control button can be pressed, mouse can be placed over it, can be activated with a particular key press
    - It is up to you to write the code (instructions) for handling each specific event you want to handle → although there are potentially many events a control can respond too, you don't have to define all of them only the ones you are interested in

Event Handling (2):

- **Control Event Handling (cont.)**
  - Basically, for each possible event, we have a separate method (function or sub-program) that will get called when the event on that control occurs
    - It is your responsibility to write these methods since the action to be performed in response to the event is program specific!
    - However, VB makes the task very easy for you → each of the potential events has an associated method and its simply a matter of you filling in the missing code

Event Handling (3):

- **Event Methods**

  >   Private Sub controlName_eventName()
  >   Intructions (code)
  >   End Sub

  - Private → Method "visibility" (ignore this for now!)
  - Sub → abbreviation for sub-program or subroutine
  - controlName → name of the control
  - eventName → event of interest (possible events are already defined by VB simply choose the desired one)
  - Instructions → the code you provide
  - End Sub → completes (ends) the method definition

Event Handling (4):

- **Event Methods (cont.)**
  - Basically a pre-defined definition of what your program should do when the user initiates the corresponding event
  - When the event occurs, the method corresponding to the event will be "called" (executed)
    - Of course, initially the method (sub-routine) will not contain any code and therefore, although it will be called, nothing will happen until you provide the code!

Event Handling (5):

- **Event Methods (cont.)**
  - So lets look at an example now
    - Control button with "click" event (e.g., user clicks on the control button with the mouse)
    - Lets assume the control button we add to the form is called "Command1" → the (empty) event method "signature" for handling the click is as follows

      >   Private Sub Command1_Click()
      >
      >   End Sub

Event Handling (6):

- **Event Methods (cont.)**
  - Once the user clicks on the Command1 button, the method "Command1_Click" will be called and any code (instructions) you have provided will be executed
   - If there are no instructions, nothing happens!

Event Handling (7):

- **Event Methods (cont.)**
  - So how do we access/write event handler methods ?
   - Double-click on the control object → code window will appear

Double-click control button

Code window
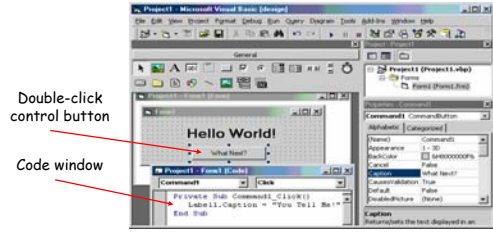
Event Handling (8):

- **Event Methods (cont.)**
  - Other available event handlers for control buttons

Particular control

Event handler methods (pre-defined but initially empty!)
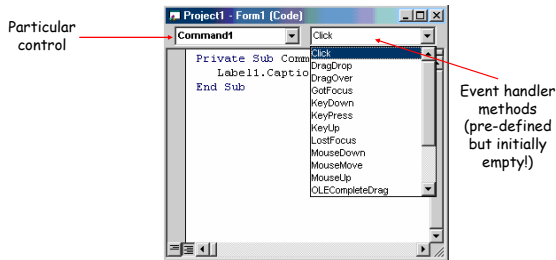
Event Handling (9):

- **Event Methods (cont.)**
  - Lets look at an actual example → "Exit" button
    - Command button name → exitButton
    - Event name → Click()
    - Form (which exitButton placed) → Form1

```
Private Sub exitButton_Click()
    Unload Form1
    End
End Sub
```


Exit button
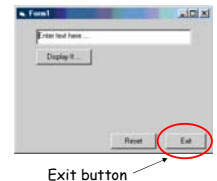
General Programming Principles (1):

- **Some VB 6.0 Syntax Rules**
  - Assignment → the "+" operator
    - The operation of giving a value to something (e.g., assigning a value to a property of an object)
  - Specifying properties of an object → the "." operator
    - The dot operator → objectName.property will identify a particular property of the object referred to by "objectName"

      Command1.Top = Label1.Top

General Programming Principles (2):

- **Some VB 6.0 Syntax Rules (cont.)**
  - Value - Examples of value "types"
    - String → sequence of characters between quotes e.g., "You Tell Me"
    - Integer → positive or negative whole numbers (no decimals) e.g., 240 & 840
    - Boolean → one of two values, either True or False

General Programming Principles (3):

- **Some VB 6.0 Syntax Rules (cont.)**
  - Names or words used in the code (two types)
    1. Word used to represent something → value, an object or a definition of something - some of these you define yourself and others are already defined for you → Form_Load, Command1_Click, Command1_MouseDown
    2. "Reserved" words that are part of the language → Private, Sub, End among many more – you cannot use any such words yourself!

General Programming Principles (4):

- **Some VB 6.0 Syntax Rules (cont.)**
  - The underscore character "_"
    - Used to continue a long statement on the next line (e.g., a line whose length will exceed the width of the code window)
    - In VB, the end of the line (instruction) is specified by the "carriage return" character (e.g., after you press "Return") as opposed to many other languages that explicitly end a line with a particular character such as a semi-colon ";"

"Live Demo" (1):

- **Live Demonstration of the Concepts Just Described Will Now be Given**
  - Control properties
    - Positioning properties (e.g., "Top")
    - Visibility property (e.g., "Visible")
  - Event methods
  - Exercise 2-2
  - Review of Exercise 2-3
  - Ask if you have any specific questions!