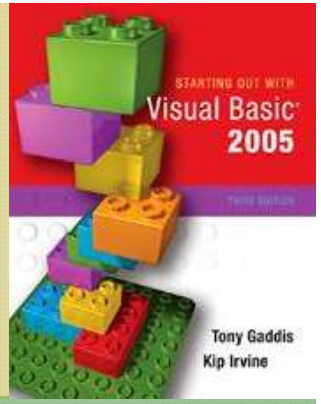
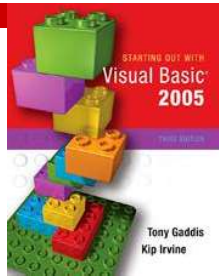


9.3 The PrintDocument Control

The PrintDocument Control Allows You to Print Data to the Printer

PrintDocument Control

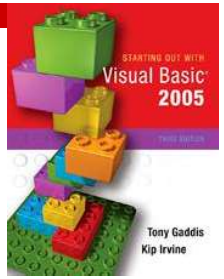


- Allows you to send output to the printer
- To use the *PrintDocument* control
  - Double click the tool in the Toolbox
  - Appears in component tray
  - Use *pd* as standard prefix when naming
- PrintDocument control has a *Print* method
  - This method starts the printing process
  - Format is:

```
PrintDocumentControl.Print()
```

  - This triggers a *PrintPage* event

PrintPage Event Handler



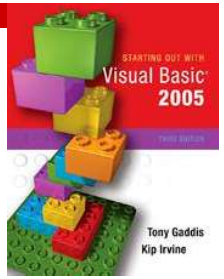
- The code in the *PrintPage* event handler performs the actual printing
  - Double click PrintDocument control in tray
  - This creates the PrintPage event handler
  - Insert your print code inside event handler
  - Basic format of event handler is as follows:

```
Private Sub pdPrint_PrintPage(ByVal sender As System.Object, _
        ByVal e As System.Drawing.Printing.PrintPageEventArgs) _
        Handles pdPrint.PrintPage

        'Your print code here

End Sub
```

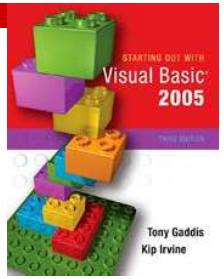
DrawString Method



- The *DrawString* method is used inside the *PrintPage* event to:
  - Specify data to send to the printer in ***string***
  - Set font, font size, and font style
  - Determine horizontal position (***HPos***) of text
  - Determine vertical position (***VPos***) of text
- DrawString method is formatted as follows:

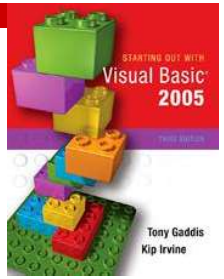
```
e.Graphics.DrawString(String, _
        New Font(FontName, Size, Style), _
        Brushes.Black, HPos, VPos)
```

Specifying Fonts, Sizes, Styles



- Fonts are specified with the string which names the font to be used
  - "Times New Roman"
- Sizes are specified with a number
  - 12
- Styles are specified with provided constants
  - FontStyle.Regular
  - FontStyle.Bold
  - FontStyle.Underline

Sample PrintPage Event Procedure

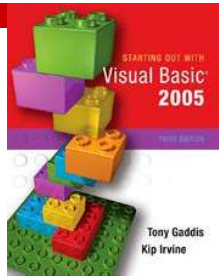



```
Private Sub pdPrint_PrintPage(ByVal sender As System.Object, _
        ByVal e As System.Drawing.Printing.PrintPageEventArgs) _
        Handles pdPrint.PrintPage
Dim inputFile As System.IO.StreamReader
Dim x As Integer = 10                    'Horizontal Position
Dim y As Integer = 10                    'Vertical Position

inputFile = System.IO.File.OpenText(filename)
Do While inputFile.Peek <> -1
        e.Graphics.DrawString(inputFile.ReadLine, _
                New Font("Courier", 10, FontStyle.Regular), _
                Brushes.Black, x, y)
        y += 12                          'Increment Vert Pos
Loop
inputFile.Close()
End Sub
```

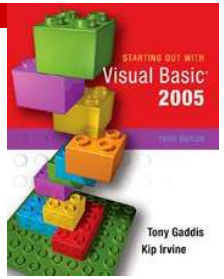
- Tutorial 9-5 adds a print feature to Tutorial 9-4

Printing Column Based Reports



- Business reports typically contain a:
  - Report header printed at the top of the page
  - Report body with the data, usually in columns
  - Optional footer, often totalling certain columns
- Report header usually has column headings
- Monospaced font used for column reports
  - Each character takes same amount of space
  - This allows columns to be aligned
- *String.Format* used to align data along column boundaries

String.Format Example




```
String.Format("{0, 10}{1, 10}{2, 10}", 50, "Arg 1", 6)
```

Specifies the argument number

Specifies field width for arg
negative - left justified
positive - right justified

Argument 0

Argument 1

Argument 2

Results in the following output:

```
        50      Arg 1         6
```

10 spaces | 10 spaces | 10 spaces