# Chapter 2 – 1<sup>st</sup> Application

# **2.1 Becoming Familiar with the Integrated Development Environment**

Visual Basic .NET provides a complete environment to assist developers in creating their applications.

This environment may vary slightly from computer to computer.

The majority of the key items are essentially the same.

The first step in creating an application with Visual Basic .NET is to familiarize yourself with the main components of the new integrated development environment.

### **Get Started page**

**Step 1.** When Visual Basic .NET is started, the **"Get Started" window** for Visual Studio .NET, similar to a home page of a Web site on the Internet, appears. This page gives you access to the following:

- Recently used projects
- > The ability to open projects not listed in the recent projects
- > An option to create new projects
- $\succ$  Links to other resources



### My Profile page

**Step 2.** The very first time you run Visual Basic .NET, you will be presented with the **My Profile screen** which will allow you to personalize the development environment settings. Once the features are set, the screen will no longer appear. Select My Profile from the VS Start Page to revisit it.



#### **New Project window**

Step 3. If you select New Project from the VS Start Page, the New Project window will appear.

For now, stick to a Windows application. Make sure Visual Basic Projects is selected from the Project Types and Windows Application is selected from the Templates.

Specify a name and place to store your new project.

New Project				
Project Types:		Templates:		00 6-6- 6-6- 6-6- 6-6-
Visual Basic Pro	j <mark>ects</mark> cts iects	VB	₩ <sup>V</sup> B	
	oyment Projects	Application	Class Library	Control Library
i Visual Studio So	olutions		E.	€ VB
		ASP.NET Web Application	ASP.NET Web Service	Web Control Library
A project for creating a	n application with a Windows	user interface		
Name:	WindowsApplication 1			
Location:	C:\VB Net Coach		•	Browse
Project will be created a	t C:\VB Net Coach\WindowsA	opplication 1.		
<b>▼</b> Mor <u>e</u>		ОК	Cancel	Help

#### **Integrated Development Environment**

Once the application has been created, you will see the IDE.

🄊 Wi	indowsApplication1 - Microsoft Visual Basic .NET [design] - Form1.vb [Design]	
Eile	Edit <u>Vi</u> ew <u>Project</u> <u>B</u> uild <u>D</u> ebug D <u>a</u> ta <u>T</u> ools <u>Wi</u> ndow <u>H</u> elp	
<b>۳</b> -	• 🏧 • 🚔 📓 👗 🖺 💼 🗠 • · · · · ·	· 🕾 🎸 ਯੋ • -
Ĭ‡	『『 キ 릐   雨 st ==   昂 拱 卯 昭   ∞ 쐈 쐈 맜 맜   옴 於 計 라   硱 凰   ृ ■ ■ .	
	Start Page Form1.vb [Design]	↓ ▷ ×         Solution Explorer - WindowsApplication1
	Form1	Solution 'WindowsApplication1' (1 pro Solution 'WindowsApplication1' (1 pro WindowsApplication1 References Sectors AssemblyInfo.vb Form1.vb
		Solution Explorer
		Properties
1	1	Form1.vb File Properties
		Build Action Compile
		Custom Tool
		Custom Tool Namesp
		Eile Name Form1.vb
		Full Path C:\VB Net Coach
		Advanced
		Advanced

#### **Form Window**

The **Form window** is like a painter's canvas. It is where you will lay out the design of a form for your application and the interface for placing the code associated with the components.



#### **Solution Explorer**

The **solution explorer window** wraps all of the components of an application into one interface. Here you can access all of the forms, projects, and other modules that are combined to build your application.

Solution Explorer - WindowsApplication 1 🛛 🕂 🗙
Solution 'WindowsApplication1' (1 project)
🕀 📾 References
E Form1.vb
Solution Explorer
Solution Explorer Class View

### **Standard Toolbar**

Visual Basic has a main or **Standard toolbar** as well as additional toolbars to give the developer easy access to commonly used operations.



#### **Toolbox Window**

Tool	box	무	×
Data	1		
Com	ponents		
Wind	dows Forms		
	Pointer		
à	Label		
Ā	Laber .		
_ <u>A</u>	LINKLADEI		
ab	Button		
abl	TextBox		
	MainMenu		
	CheckBox		
$\odot$	RadioButton		
[×v]	GroupBox		
	PictureBox		
m	Panel		
5	DataGrid		
-	ListBox		
	CheckedListBox		
	ComboBox		
222	ListView		
18-	TreeView		
1	TabControl		
	DateTimePicker		
	MonthCalendar		
REF	HScrollBar		
	VScrollBar		
之	Timer		
+++	Splitter		
E A	Domain! InDown		
13. 13.	NumericUpDown		
,	TrackBar		
	ProgressBar		
	RichTextBox		
	ImageList		
	HelpProvider		
	TeelTie		
Net I	Controllanu		
5	Contextmenu		
	ChabusBas		
FE	Statusbar		
ND2	NotityIcon		
	OpenFileDialog		
1 1 1	SaveFileDialog		
A	FontDialog		
<u>•</u>	ColorDialog		
	PrintDialog		
	PrintPreviewControl		
	ErrorProvider		
	PrintDocument		
	Princocument		
	CrystalReport//icure		
Clipt	oard Ring		-
Gen	eral		

**Toolbox window** contains objects called controls that assist in creating applications. The controls are broken into sections.

#### **Properties Window**

Components of Visual Basic have many properties associated with them. A **property** is a way of customizing the appearance and behavior of a control in Visual Basic .NET.

Amongst other properties, the **Properties window** allows the developer to control the color, font, and size of Visual Basic .NET constructs. The current property selected is highlighted, and additional information explaining the purpose of that property is displayed at the bottom of the window. Properties are sometimes grouped together.

•	2 2↓ 🔲 🖻		
Ξ	Accessibility		
	AccessibleDescriptio		
	AccessibleName		
	AccessibleRole	Default	
Ξ	Appearance		
	BackColor	Control 🗨	
	BackgroundImage	(none)	
	Cursor	Default	
Ŧ	Font	Microsoft Sans Serif, 8	
	ForeColor	ControlText	
	FormBorderStyle	Sizable	
	RightToLeft	No	
	Text	Form1	
Ξ	Behavior		
	AllowDrop	False	
	ContextMenu	(none)	
	Enabled	True	
	ImeMode	NoControl	
Ξ	Configurations		
Ŧ	(DynamicProperties)		
Ξ	Data		
Ŧ	(DataBindings)		
	Tag		
Bi Th gr	ackColor ne background color u aphics in the control.	sed to display text and	

### Menu Bar

The Visual Basic .NET **Menu bar** is located just below the Title bar. It contains shortcuts to commonly used commands.

<u>File Edit View Project Build Debug Data Tools Window Help</u>

### **2.2 Creating Your First Application**

For your first application, you are going to create a Form that acts out the classic tale of "The Lady or the Tiger" by Frank R. Stockton. You will develop this application in a stepwise fashion.







#### Setting the Project Name and Form Name

**Step 1:** From the Start window, click on New Project. The New Project window will appear.

New Project				
Project Types:		Templates:		
Visual Basic Proj Visual C# Proj Visual C++ Pro Setup and Dep Other Projects Visual Studio S	ojects ojects oloyment Projects s olutions	Windows Application Application ASP.NET Web Application	Class Library Class Library ASP.NET Web Service	Windows Control Library
A project for creating	an application with a Windows	user interface		
Name:	Lady or the Tiger			
Location:	C:\VB Net Coach\Chapter 2	\Code	•	Browse
Project will be created	at C:\VB Net Coach\Chapter 2	Code\Lady or t	ne Tiger.	
<b>▼</b> Mor <u>e</u>		ОК	Cancel	Help

Step 2: Specify the name of the application as "Lady or the Tiger".

Step 3: Specify the location as "C:\VB Net Coach\Chapter 2\Code\".

Step 4: Click on the OK button and the development environment will appear.



#### Setting the Form Name

**Step 5:** You should rename the default form of the application Form1 to something more meaningful. Visual Basic .NET allows you to set a value to display as the name for the form, to set the name you refer to the form from within your program, and to set the actual file name.

When naming Forms in Visual Basic, your name should follow Microsoft's naming convention. The prefix identifies what type of object you are naming. In this case, use the prefix frm to indicate that your name is associated with a Form object.

Double-click on the window containing the name Form1 to highlight it.

Pr	operties	<b>₽ X</b>
F	orm1 System.Windo	ws.Forms.Form
-	. 2↓ 🔲 📼	
Ξ	Design	<b>▲</b>
	(Name)	Form1
	DrawGrid	True
Ŧ	GridSize	8, 8
	Locked	False
	SnapToGrid	True
Ξ	Focus	
	CausesValidation	True
Ξ	Layout	
	AutoScale	True
	AutoScroll	False
Ŧ	AutoScrollMargin	0,0
Ŧ	AutoScrollMinSize	0,0
Ŧ	DockPadding	
Ŧ	Location	0, 0
Ŧ	MaximumSize	0, 0
Ŧ	MinimumSize	0,0
Ŧ	Size	300, 300
	StartPosition	WindowsDefaultLocati
	WindowState	Normal
Ξ	Misc	
	AcceptButton	(none)
	CancelButton	(none) 👻
(1	lame)	

#### **Setting the Form Name**

Step 6: Type the name for the form, frmLadyOrTiger in the Properties window.

Pr	operties	<b>д X</b>
fr	mLadyOrTiger Sy	stem.Windows.Forms.F 🔻
ŀ	1 🛃 🔲	
Ξ	Design	
	(Name)	frmLadyOrTiger
	DrawGrid	True
Ŧ	GridSize	8, 8
	Locked	False
	SnapToGrid	True
Ξ	Focus	
	CausesValidation	True
Ξ	Layout	
	AutoScale	True
	AutoScroll	False
Ŧ	AutoScrollMargin	0,0
Ŧ	AutoScrollMinSize	0,0
Ŧ	DockPadding	
Ŧ	Location	0,0
Ŧ	MaximumSize	0,0
Ŧ	MinimumSize	0,0
Ð	Size	300, 300
	StartPosition	WindowsDefaultLocati
	WindowState	Normal
Ξ	Misc	
	AcceptButton	(none)
	CancelButton	(none)
(1	Name)	

#### **Setting the Form Name**

**Step 7**: You can rename the file by right-clicking on the file name in the Solution Explorer. A pop-up window will appear that will allow you to select Rename.



Step 8: Enter the file name for the form, LadyOrTiger.vb

#### **Setting the Form Text**

Step 9: To set the display name for the form set the Text property to Lady Or The Tiger.

Pr	operties	д	×
fr	mLadyOrTiger Sys	tem.Windows.Forms.F	•
	. 2↓ 🔲 🖻		
⊡	Accessibility		٠
	AccessibleDescription		
	AccessibleName		
	AccessibleRole	Default	
⊡	Appearance		
	BackColor	Control	
	BackgroundImage	(none)	
	Cursor	Default	
Ð	Font	Microsoft Sans Serif, 8	
	ForeColor	ControlText	
	FormBorderStyle	Sizable	
	RightToLeft	No	
	Text	Lady or the Tiger	
Ξ	Behavior		
	AllowDrop	False	
	ContextMenu	(none)	
	Enabled	True	
	ImeMode	NoControl	
Ξ	Configurations		
Ŧ	(DynamicProperties)		
⊡	Data		
Ŧ	(DataBindings)		
	Tag		Ŧ
Te Th	e <b>xt</b> ne text contained in th	e control.	

#### **Placing a Label Control on the Form**



A label control allows you to add static text to a form that will typically not change during the running of your application.

**Step 1**: Select the label control by clicking on the **icon** for the Label control .

#### **Placing the Label Control**

Step 2: Place the label in the desired position on the form.



#### **Placing the Label Control**

**Step 3**: Release the mouse button, thereby completely specifying the location, width, and height of the text box.

Start Page	Form1.vb [Design]*	
🔡 Lady	or the Tiger	
Label1		
	a na provinski provin	

#### **Placing the Label Control**

**Step 4:** Make sure that your Label control is roughly the same size as shown. If it is not, there are two ways that you can modify your control.

You can adjust it visually.

You can adjust it by modifying properties of the Label control. The Width and Height properties can be set to the exact values. Set yours to 248, 40, respectively.

Step 5: Set the Name property of the Label to lblTitle.

Step 6: Click the Text property and type Pick a Door Wisely.



### **Changing the Label Size and Style**

**Step 7:** We can set a font's type face, size, and style.

Set the Font.Style to Bold and the Size to 18.

Properties	<b>₽ X</b>	Font			? 🗙	Properties	4
frmLadyOrTige	r System.Windows 🔻	Font:	Font style:	Size:		IbITitle System.W	/indows.Forms.Label
B 🛃 🔲 🗉	3	Microsoft Sans Serif	Regular	8	ОК	∄ 2↓ 🗉 🖻	
	_	🥖 Microsoft Sans Serif 🔥	Regular	8 🔨	Cancel	Accessibility	
AccessibleDesr		T Monotype Corsiva	Bold	10		AccessibleDescri	ptio
AccessibleNam		Tr MS Outlook	Bold Italic	11		AccessibleName	
AccessibleRole	Default	Thews Gothic		14		AccessibleRole	Default
	Derduit	"In News Gothic Condens ≥		16 💌		Appearance	
BackColor	Control	- Effects	- Sample			BackColor	Control
BackgroundIm		Strikeout				BorderStyle	None
Cursor			AaBbYyZ	Zz		Cursor	Default
E East	Microsoft Sana	) ondenine	1			FlatStyle	Standard
Font			Script:			Font	Microsoft Sans Ser
ForeColor	La Sizable		Western	•		Name	ab Microsoft San
PormborderSty	le Sizable					Size	8.25
RightToLeft						Unit	Point
Text	Lady or the Tig					Bold	False
Behavior						GdiCharSet	0
AllowDrop	False					GdiVerticalFor	nt False
ContextMenu	(none)					Italic	False
Enabled	True					Strikeout	False
ImeMode	NoControl					Underline	False
Configuration	IS					ForeColor	ControlText
① (DynamicPrope	rt					Image	(none)
🗆 Data	-					ImageAlign	MiddleCenter
-						ImageIndex	(none)

### **Aligning the Label Text**

**Step 8:** The final step is to align the text so that it is centered in the form.

This can be done visually or by clicking on the <code>TextAlign</code> property and selecting <code>MiddleCenter</code> from the pull-down menu.



### **2.3 Picture Box Control**

Visual Basic allows you to easily add pictures to your form. In your case, you wish to add a picture of a door twice. To display a picture, you will use the **picture box control**.



### **Select PictureBox Control**

**Step 1:** Start by clicking on the PictureBox control in the control toolbox.

### **Placing the PictureBox Control**

Step 2: Click just below and to the left of the text in the Label control you previously placed.

Hold the mouse button down and release it with the mouse pointer near the bottom of the form and aligned in between the o's of the word "Door" in the label control.

🔜 Lady	or the Tiger			
Р	Pick a Door Wisely			
	·····			

#### **Setting the Name of the Control**

Step 3: Set the Name property of the PictureBox to picLeftDoor.

### **Setting the Picture to Display in the Control**

Step 4: Click on the Image property

Click on the ellipses (...) and a dialog box will appear to select the graphic file to display within the PictureBox control.



#### **Resizing the Picture Box Control**

**Step 5:** Since the form is not large enough, increase its size by setting the Width and Height to 624, 424, respectively. Click anywhere on the form a control is not already placed to select the form and display its properties.

Increase the size of the PictureBox by setting the Height and Width properties to 100, 248, respectively.



#### **Add Other Door**

**Step 6:** The last step is to repeat the process and create another Picture Box, picRightDoor, with the same picture.



### **2.4 Text Box Control**

By using a **text box control** you can place an area on the form where users of the application may enter any text he or she wishes.



### **Select the Text Box From the Control Toolbox**

Step 1: Select the TextBox control from the control toolbox.

#### Add Text Box to Form

Step 2: Place a text box on the form in the same manner as the other controls.



#### Setting the Name of the Control

Step 3: Set the Name property of the text box to txtName .

#### **Clearing the Default Text**

Step 4: Click on the Text property and erase "TextBox1".

🖬 Lady or the Tiger		
	Pick a Door Wisely	



#### Adding a Label above the Text Box

**Step 5:** Add a Label control above the text box to indicate the text box is for the peasant's name. Set the Name of the label to lblName, the Font Bold to True and the Text to Name.



### Chapter 2 – 1<sup>st</sup> Application

### **2.5 Button Control**



### **Select the Button Control From the Toolbox**

**Step 1:** Select the Button control from the control toolbox.

#### **Add Command Button to Form**

Step 2: Place a Button on the form in the same manner as the other controls.



#### **Setting the Name of the Control**

Step 3: Set the Name property to btnLeftDoor .

#### **Clearing the Default Text**

Step 4: Set the Text property to "Left Door".

Pick a Door Wisely     Image: Constraint of the second	🔜 Lady or the Tiger		
Pick a Door Wisely         Image: State of the state		• •	
Name     Left Door		Diek e Deer Wieely	
Name	• • • • • • • • • • • • • • • • • • • •	PICK a DOOL WISELY	
Name         Left Door		**	
Name			
Name	• • • • • • • • • •		
Name			
Name Left Door			
Name       Image: Second			
Name			
Name         Image: State of the state			· · · · · · · · · · · · · · · · · · ·
Name       Left Door			
Left Door		Name	
Left Door			· ****
Left Door			
Left Door			·
Left Door	and a second sec		bisister discounts
Left Door			
Left Door	••••••		·
Left Door			
Left Door			
	Left Door		
		4	
	• • • • • • • • • • • • • • • • • • • •		
	• • • • • • • • • • • • • • • • • • • •		
• • • • • • • • • • • • • • • • • • • •			

#### Add the Other Button

Step 5: The last step is to repeat the process and create another Button, btnRightDoor.



### 2.6 Basic Event Handling

You now will add functionality so that clicking on your button will cause the form to display either a young maiden or a tiger.

If the young maiden is displayed, the label should change to the person's name and the words " is Innocent". If the tiger appears, the label should change to the person's name and " is Guilty".

You will also notice whatever text is in the text box is erased.

This involves your first real coding. In this case, your coding will be triggered by an event.

### **Display the Code for the Button**

Step 1: Double-click on the btnLeftDoor button to display the code.

Start Page frmLadyOrTiger.vb [Design]* frmLadyOrTiger.vb*	4 ◊ ×
∲¢ frmLadyOrTiger	·
Public Class frmLadyOrTiger     Inherits System.Windows.Forms.Form	
+ Windows Form Designer generated code	
Private Sub btnLeftDoor_Click(ByVal sender As System.Object	;, ByVal e As System.EventArgs) Handles btnLeftDoor.Click
- End Sub End Class	

The first few lines define the object you are working in. In this example you are creating a form called frmLadyOrTiger.

The first two words, Public Class, indicate that you are creating a template for an object that can be accessed by any object in the project.

The code for the event starts with the words Private Sub

btnLeftDoor\_Click() indicates that this code is attached to the btnLeftDoor button and will be exected when a Click event occurs.

End Sub indicates the ending of the event.

#### **Clearing the Text Box**

**Step 2:** Remove any text that is placed in the txtName text box by setting the Text property.

You can use the pull-down menu to select the property you wish to set or type it in.

frmLadyOrTiger		•	atthe btnLeftDoor_Click	
- Public Class	frmLadvOrTiger			
Inherits	System.Windows.Forms.Fo	rm		
	-			
H Windows For	m Designer generated cod	le		
- Private	Sub btnLeftDoor Click(By	Val sender As System.Object	, ByVal e As System.EventArgs) Handles btnLeftDoor.Click	
txtN	lame.Te	1	1 1 3,	
- End Sub	🖦 SuspendLayout			
-End Class	rest TabIndex			
	tabstop tabstop			
	Text			
	™ TextAlign			
	ToString			
	and ToString and Undo			
	a Totking ⊕ Undo ▼			
	a ToString a ToString a Undo ▼			
	a fostring ● Tostring ● Undo			4
tart Page   frmLadyOrTiger.	vb [Design]* frmLadyOrTiger.vb*		-@html.eftDoor.click	4
tart Page   frmLadyOrTiger. <b>{ frmLadyOrTiger</b>	vb [Design]* frmLadyOrTiger.vb*			4
tart Page   fmLadyOrTiger. \$fmLadyOrTiger □ Tublic Class	vb [Design]* frmLadyOrTiger.vb*	•		4
tart Page   frmLadyOrTiger. frmLadyOrTiger Public Class Inherit:	vb [Design]* frmLadyOrTiger.vb*	• •rm	ja∯btnLeftDoor_Click	4
tartPage fmLadyOrTiger fmLadyOrTiger Public Class Inherits Windows Fo:	vb [Design]* frmLadyOrTiger.vb*	• orm le	ja∯btnLeftDoor_Click	4
startPage   fmLadyOrTiger. \$fmLadyOrTiger □ Public Class Inherits + Windows For	vb [Design]* frmLadyOrTiger.vb*	• orm le	ja <sup>®</sup> btnLeftDoor_Click	4
tart Page   fmLadyOrTiger. fmLadyOrTiger Public Class Inherits Windows For	<pre>vb [Design]* frmLadyOrTiger.vb*   s frmLadyOrTiger s System.Windows.Forms.Fo cm Designer generated cod</pre>	orm le	ja <sup>®</sup> btnLeftDoor_Click	4
tart Page   fmLadyOrTiger. \$fmLadyOrTiger Public Class Inherits \$\overline{1} Windows Formations Private	<pre>vb [Design]* frmLadyOrTiger.vb*   s frmLadyOrTiger s System.Windows.Forms.Fo cm Designer generated cod Sub btnLeftDoor_Click(By lame mout   </pre>	orm Ne System.Object,	ja♥btnLeftDoor_Click ByVal e As System.EventArgs) Handles btnLeftDoor.Click	4
Start Page   fmLadyOrTiger. \$fmLadyOrTiger Public Class Inherits Windows Fo: Private txtl Fod Text	vb [Design]* frmLadyOrTiger.vb*   s frmLadyOrTiger s System.Windows.Forms.Fo rm Designer generated cod Sub btnLeftDoor_Click(By Name.Text   Dassmod	orm Ne Wal sender As System.Object,	ja♥btnLeftDoor_Click ByVal e As System.EventArgs) Handles btnLeftDoor.Click	4

**Step 1 Continued:** To cause a text box's Text property to change, type an equal sign followed by the new value you want to display in quotation marks

	Start Page   frmLadyOrTiger.vb [Design]* frmLadyOrTiger.vb*		4 ⊳ ×
ſ	t frmLadyOrTiger	j ja∲btnLeftDoor_Click	•
	Public Class frmLadyOrTiger Inherits System.Windows.Forms.Form		<b>^</b>
	H Windows Form Designer generated code		
	<pre>Private Sub btnLeftDoor_Click(ByVal sender As System.Object</pre>	, ByVal e As System.EventArgs) Handles btnLeftDoor.Click	
	-End Class		

Now you can run your application by clicking on the Start button in the Standard toolbar. Type a name into the text box like "Michael" and click on the button btnLeftDoor. Watch how the text in the text box is removed.

Lady or the Tiger	Lady or the Tiger
Pick a Door Wisely	Pick a Door Wisely
Name       Michael       Left Door       Right Door	Image: Name     Image: Name       Left Door     Right Door

#### **Changing the Picture and Label**

Step 2 : Set the Image property of picLeftDoor to the new graphic.

Type Image.FromFile("Picture name and path goes here") to the right of the equal sign. In your case the picture name and path are "Tiger.gif".

Change the Label control lblTitle to "Michael is Guilty:

Start Page   frmLadyOrTiger.vb [Design]*   frmLadyOrTiger.vb*	4 ▷ ×
frmLadyOrTiger	de btnLeftDoor_Click
Public Class frmLadyOrTiger	
Inherits System.Windows.Forms.Form	
<pre>Private Sub btnLeftDoor_Click(ByVal sender As System.Object</pre>	, ByVal e As System.EventArgs) Handles btnLeftDoor.Click

In order to combine the name from txtName text box and the text "is Guilty", you must use a concatenation operation. The & in between txtName.Text and "is Guilty" will do that.

Now you can run your application. Click on the btnLeftDoor button to see the final result.

🔜 Lady or the Tiger		
	Michael is Guilty	
Left Door	Name	Right Door

#### **Code the Right Doors Button**

**Step 3 :** Set the code for the btnRightDoor command button in a similar fashion.

	Start Page	e frmLadyOrTiger.vb [Design] frmLadyOrTiger.vb	$\triangleleft \triangleright \times$
R	trmLa	dyOrTiger 🗾 🚽 🚽 dy btnRightDoor_Click	•
	- Pu	blic Class frmLadyOrTiger	
		Inherits System.Windows.Forms.Form	
	+W	indows Form Designer generated code	
		<pre>Private Sub btnLeftDoor_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnLeftDoor.Click     picLeftDoor.Image = Image.FromFile("Tiger.gif")     lblTitle.Text = txtName.Text &amp; " is Guilty"     txtName.Text() = "" Fred Sub</pre>	
	-	<pre>Private Sub btnRightDoor_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRightDoor.Click     picRightDoor.Image = Image.FromFile("Lady.gif")     lblTitle.Text = txtName.Text &amp; " is Innocent"     txtName.Text() = "" End Sub</pre>	
	- En	d Class	

Here is the final result of clicking the btnRightDoor button.

🔜 Lady or the Tiger		
	Michael is Innocent	
Left Door	Name	Right Door

### **Saving the Project**

You can save your project by either clicking on the Save icon in the Standard toolbar, or Selecting File and Save from the Menu bar.

### 2.7 Use of Color

Visual Basic .NET gives you the ability to set the color of almost every control. Your use of color should be conservative, consistent, and logical.

Imagine if you were developing a travel application.

Which do you like best?





Which application looks the best is a matter of preference. Personally, I like the second one the best.

### **Changing Color**

The two properties that were used in this example are ForeColor and BackColor. Each can be set interactively or programmatically. To set the color interactively, perform the following:

Step 1: Click on the control that you wish to change the color of.

Step 2: Click on the property you wish to change. (ForeColor or BackColor).

Step 3: Click on the drop-down arrow to get the Pallet window to appear.

**Step 4:** Click on the Custom tab of the pop-up window.

Step 5: Click on the color you wish to select:



#### **Programatically Changing Color**

We can also set colors programmatically by using any of the predefined Visual Basic .NET colors. If we wanted to set the vacation application to display Egypt instead of Ireland, we could use the following code:

\$ frmTravel	▼ Declarations)	•
🖂 Public Class frmTravel		
Inherits System.Windows.Forms	Form	
Windows Form Designer generated	code	
Чуу 		
Private Sub btnChangeDestinat	ion_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Ha	ndles btnChangeDestination.Clic
<pre>lblTitle.Text = "Egypt is</pre>	the place to be!"	
picPhoto.Image = Image.Fr	omfile("Egypt.jpg")	
txtName.ForeColor = Color	Gold	
lblTitle.ForeColor = Colo	r.Gold	
- End Sub		
- End Class		
	🖬 Travel 🔲 🗖 🔯	
	Egypt is the place to be!	
	the second s	
	and the second	
	Change Destination	
	East	
	1-31M	
		2
		-

### 2.8 Case Study

### **Problem Description**

A company that sells products on the Internet, Walking Promotions, wants to develop an application to track its financial data.

Create an application that allows the entry of the names of the employees, how many hours they worked that week, and a place to display their payment for the week as well as a total cost of payroll.

Beautify the form by adding the company's logo to the form as well as a title.

Payroll Accounting Sys	stem	
-A-	Payroll Ac	counting System
Employee Name	Hours Worked	Weekly Pay
	Tota	al Pau
	,,,,,	

### 2.8 Case Study

### **Problem Discussion**

While you do not know enough to create an application that will actually process the payroll, you can at least set up the user interface of the application.

It will require using text boxes, picture boxes, and label controls.

To make programming easier in the future, it is a good idea to name all of the controls something a little more specific than the default values.

This will make them more discernible later.

#### **Problem Solution**

The project requires a single form and a graphic file containing the logo.

The graphic file will be called WalkingPromotionsLogo.jpg.

While you have many options in how you lay out your solution, you want to come up with a simple, intuitive solution. Your completed application should look as follows:

Payroll Accounting System		
	Payroll Acco	unting System
Employee Name	Hours Worked	Weekly Pay
	Total Pag	y

#### **Adding the Logo**

To add the Walking Promotion's logo to a blank form, follow the steps that you used when adding a PictureBox control in Section 2.3.

**Step 1:** Select the PictureBox control from the control toolbar.

**Step 2:** Place the mouse pointer over the area of the form you wish to place the upper-left corner of the logo.

**Step 3:** Hold the mouse button down and drag the pointer to the lower-right corner where the logo will be placed.

Step 4: Release the mouse button and the PictureBox control will be placed on the form.

**Step 5:** Click on the Image property in the Properties window and click on the WalkingPromotionsLogo.jpg to select the appropriate graphic.

Step 6: Click on the Name property and change the Name property to picLogo.

#### **Adding the Labels**

You need to add a total of five labels to the form. Add the first four now and the last one after you add the text box controls so that they line up appropriately. To add the first control, perform the following steps:

**Step 1: Select the** Label **control from the** toolbox.

**Step 2:** Place the mouse over the area of the form you wish to place the upper-left corner of the Label control.

**Step 3:** Hold the mouse button down and drag the pointer to the lower-right corner where the Label control will be placed.

Step 4: Release the mouse button and the Label control will be placed on the form.

**Step 5: Click on the** Text **property in the** Properties window **and type** "Payroll Account System".

Step 6: Click on the Font property in the Properties window and set the Size to 14 and the Bold property to True.

Step 7: Click on the Name property and change the Name property to lblTitle.

#### **Adding the Labels Continued**

The following are the properties for the other three label controls you need to add to the form:

Name: IblEmployeeName	Name: IblHoursWorked	Name: IblWeeklyPay
Text: Employee Name	Text: Hours Worked	Text: Weekly Pay
Font: Size=11, Bold=True	Font: Size=11, Bold=True	Font: Size=11, Bold=True
Size: Width=128, Height=23	Size: Width=112, Height=23	Size: Width=104, Height=23
Location: X=16, Y=112	Location: X=240, Y=112	Location: X=376, Y=112

#### **Adding the Text Boxes**

You need to add a total of 13 text box controls to the form. Remember to line them up when you are finished. To add the first text box perform the following steps:

**Step 1:** Select the TextBox control from the control toolbar.

**Step 2:** Place the mouse over the area of the form you wish to place the upper-left corner of the TextBox control.

**Step 3:** Hold the mouse button down and drag the pointer to the lower-right corner of where the TextBox control will be placed.

Step 4: Release the mouse button and the TextBox control will be placed on the form.

Step 5: Click on the Name property in the Properties window and type txtEmployee1.

**Step 6:** Click on the Text property in the Properties window and clear the default text so that nothing is displayed in the text box when you run the application.

### **Adding the Text Boxes Continued**

The following are the properties for the remaining text box controls you need to add to the form:

Name: txtEmployee2	Name: txtEmployee3	Name: txtEmployee4
Text:	Text:	Text:
Size: Width=192, Height=20	Size: Width=192, Height=20	Size: Width=192, Height=20
Location: X=16, Y=168	Location: X=16, Y=200	Location: X=16, Y=232
Name: txtHours1	Name: txtHours2	Name: txtHours3
Text:	Text:	Text:
Size: Width=80, Height=20	Size: Width=80, Height=20	Size: Width=80, Height=20
Location: X=240, Y=136	Location: X=240, Y=168	Location: X=240, Y=200
Name: txtHours4	Name: txtWeeklyPay1	Name: txtWeeklyPay2
Name: txtHours4 Text:	Name: txtWeeklyPay1 Text:	Name: txtWeeklyPay2 Text:
Name: txtHours4 Text: Size: Width=80, Height=20	Name: txtWeeklyPay1 Text: Size: Width=112, Height=20	Name: txtWeeklyPay2 Text: Size: Width=112, Height=20
Name: txtHours4 Text: Size: Width=80, Height=20 Location: X=240, Y=232	Name: txtWeeklyPay1 Text: Size: Width=112, Height=20 Location: X=376, Y=136	Name: txtWeeklyPay2 Text: Size: Width=112, Height=20 Location: X=376, Y=168
Name: txtHours4 Text: Size: Width=80, Height=20 Location: X=240, Y=232 Name: txtWeeklyPay3	Name: txtWeeklyPay1 Text: Size: Width=112, Height=20 Location: X=376, Y=136 Name: txtWeeklyPay4	Name: txtWeeklyPay2 Text: Size: Width=112, Height=20 Location: X=376, Y=168 Name: txtTotalPay
Name: txtHours4 Text: Size: Width=80, Height=20 Location: X=240, Y=232 Name: txtWeeklyPay3 Text:	Name: txtWeeklyPay1 Text: Size: Width=112, Height=20 Location: X=376, Y=136 Name: txtWeeklyPay4 Text:	Name: txtWeeklyPay2 Text: Size: Width=112, Height=20 Location: X=376, Y=168 Name: txtTotalPay Text:
Name: txtHours4 Text: Size: Width=80, Height=20 Location: X=240, Y=232 Name: txtWeeklyPay3 Text: Size: Width=112, Height=20	Name: txtWeeklyPay1 Text: Size: Width=112, Height=20 Location: X=376, Y=136 Name: txtWeeklyPay4 Text: Size: Width=112, Height=20	Name: txtWeeklyPay2 Text: Size: Width=112, Height=20 Location: X=376, Y=168 Name: txtTotalPay Text: Size: Width=112, Height=20

#### **Adding the Final Label**

The following are the properties for last label control you need to add to the form:

Name: lblTotalPay

Text: Total Pay

Font: Size=11, Bold=True

Size: Width=72, Height=23

Location: X=296, Y=264

### **Coach's Corner**

### **Tab Index**

When you create applications, you often add controls in the order you think about them, not necessarily in the order the user wishes to enter data.

When you place controls on a form, they receive a default TabIndex that follows the order they are added to the form.

A well-designed program should start with the focus in the upper-left corner. When the <TAB> key is pressed, you should move to the next logical control.

You can control the tab index of the form by setting the TabIndex property of each control to the logical sequence starting with 0.

### **Tab Stop**

Often you do not wish a control to be in the tab order at all.

You should set the TabStop property of the control to False.

A False setting will remove the control from the tab loop.

A True setting will place it in the tab loop. PictureBox and Label controls usually do not have a tab order.