CHAPTER 1: INTRODUCTION TO VISUAL BASIC

- Describe steps for developing a Visual Basic Application
- Use textbox, button, checkbox, label, picturebox, radiobutton.
- Explaining about controls and programming
- Identify compile errors, run-time errors and logic errors.

INTRODUCING MICROSOFT VISUAL BASIC

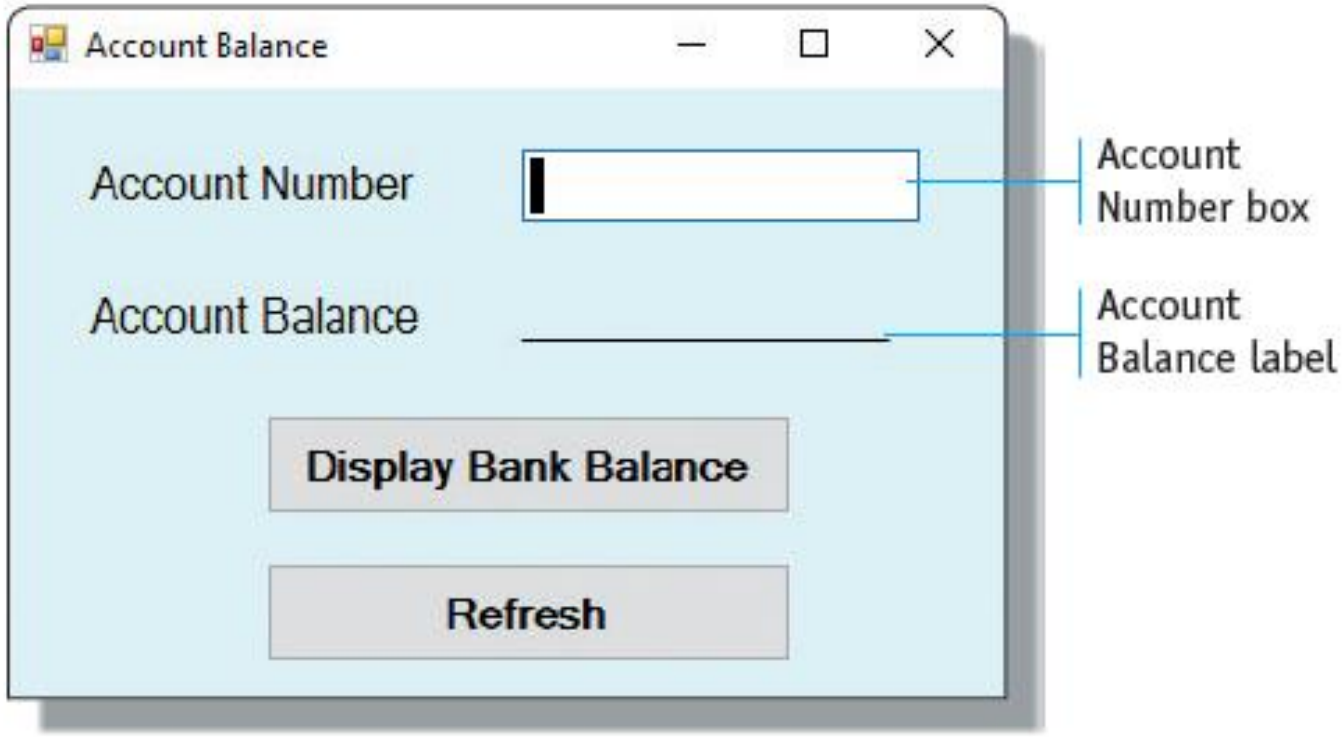
- **What is Visual Basic?**
  - Programming language that allows developers to easily build complex Windows and web programs, as well as other software tools
  - Based on the BASIC language
  - Commonly referred to as VB
- **What can you do with Visual Basic?**
  - Create applications with graphical windows, dialog boxes, and menus
  - Create applications that work with databases
  - Create Web applications and applications that use Internet technologies
  - Create applications that display graphics

EVENT-DRIVEN COMPUTER PROGRAMS WITH A GRAPHICAL USER INTERFACE (1 OF 2)

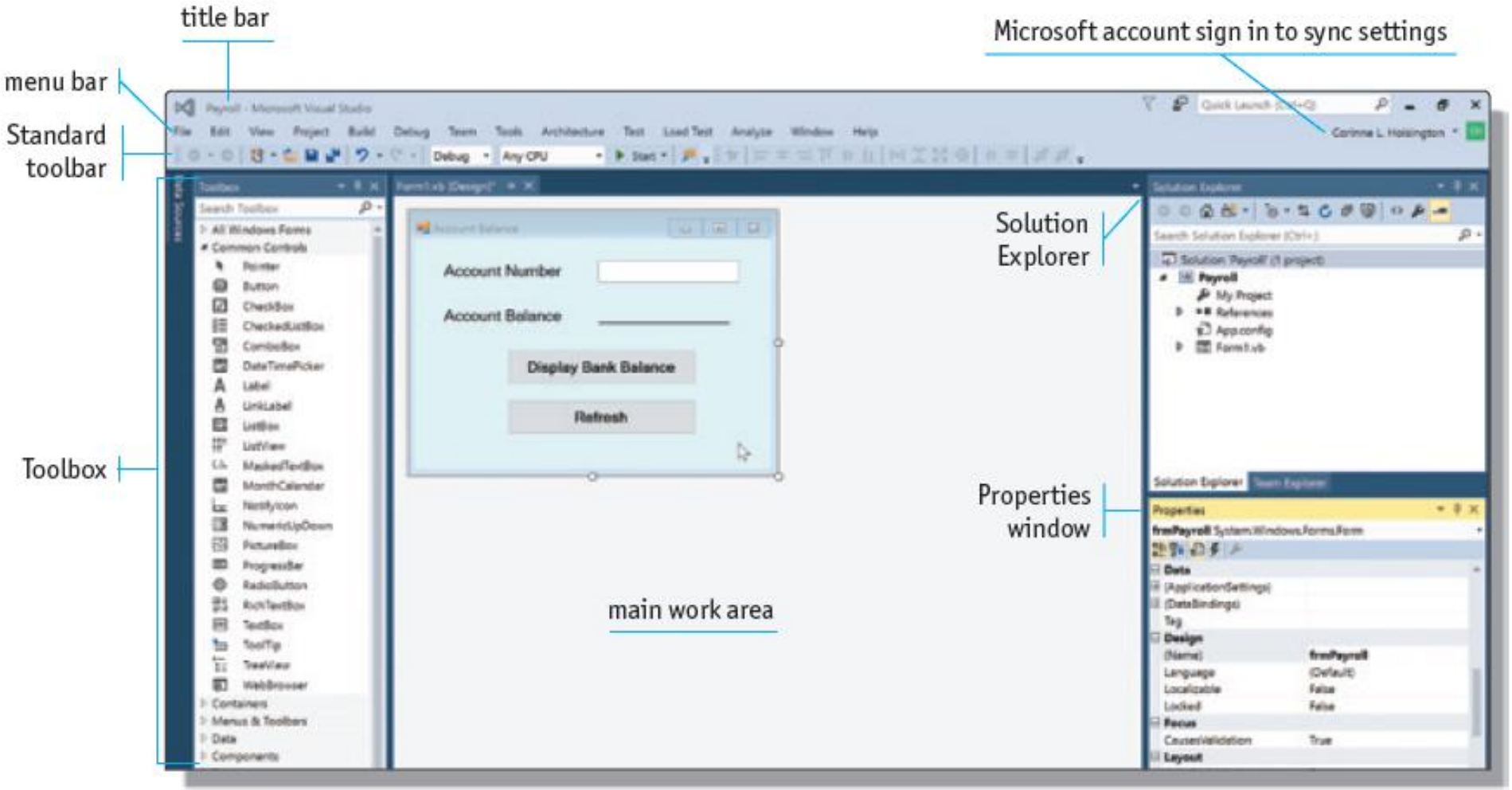
- Most Visual Basic 2017 programs are **event-driven programs** that communicate with the user through a **graphical user interface** (**GUI**)
  - The GUI usually consists of a window, containing a variety of objects
  - Defines how elements look and function
- An **event** means the user has initiated an action that causes the program to perform a type of processing in response to that action

EVENT-DRIVEN COMPUTER PROGRAMS WITH A GRAPHICAL USER INTERFACE (2 OF 2)

- For example:
  - A user might enter data into the program and then click a button



VISUAL STUDIO 2017 WINDOW (1 OF 3)



VISUAL STUDIO 2017 WINDOW (2 OF 3)

- The following elements help you to use the Visual Studio 2017 Window:
  - **Title bar:** The title bar identifies the window and the application open in the window
  - **Menu bar:** The menus contain lists of commands that allow you to create, edit, save, print, test, and run a Visual Basic program and to perform other functions
  - **Standard tool bar:** The Standard tool bar contains buttons that execute frequently used commands such as Open Project, New Project, Save, Cut, Copy, Paste, and Undo
  - **Toolbox:** The **toolbox** contains **.NET components** that help you develop the GUI program
  - **Main work area:** The main work area contains the item you are currently developing
  - **Solution Explorer:** The Solution Explorer window displays the elements of the Visual Basic **solution**

OBJECTS MODEL

- Visual Basic is an object-oriented programming (OOP) language
- **Object** is an item in a program that contains data and has the ability to perform actions
- The data an object contains is referred to as **properties**, or attributes
- The operations that an object can perform are called **methods**
- **Events** occur when the user takes action
- **Classes** are templates used to create a new object

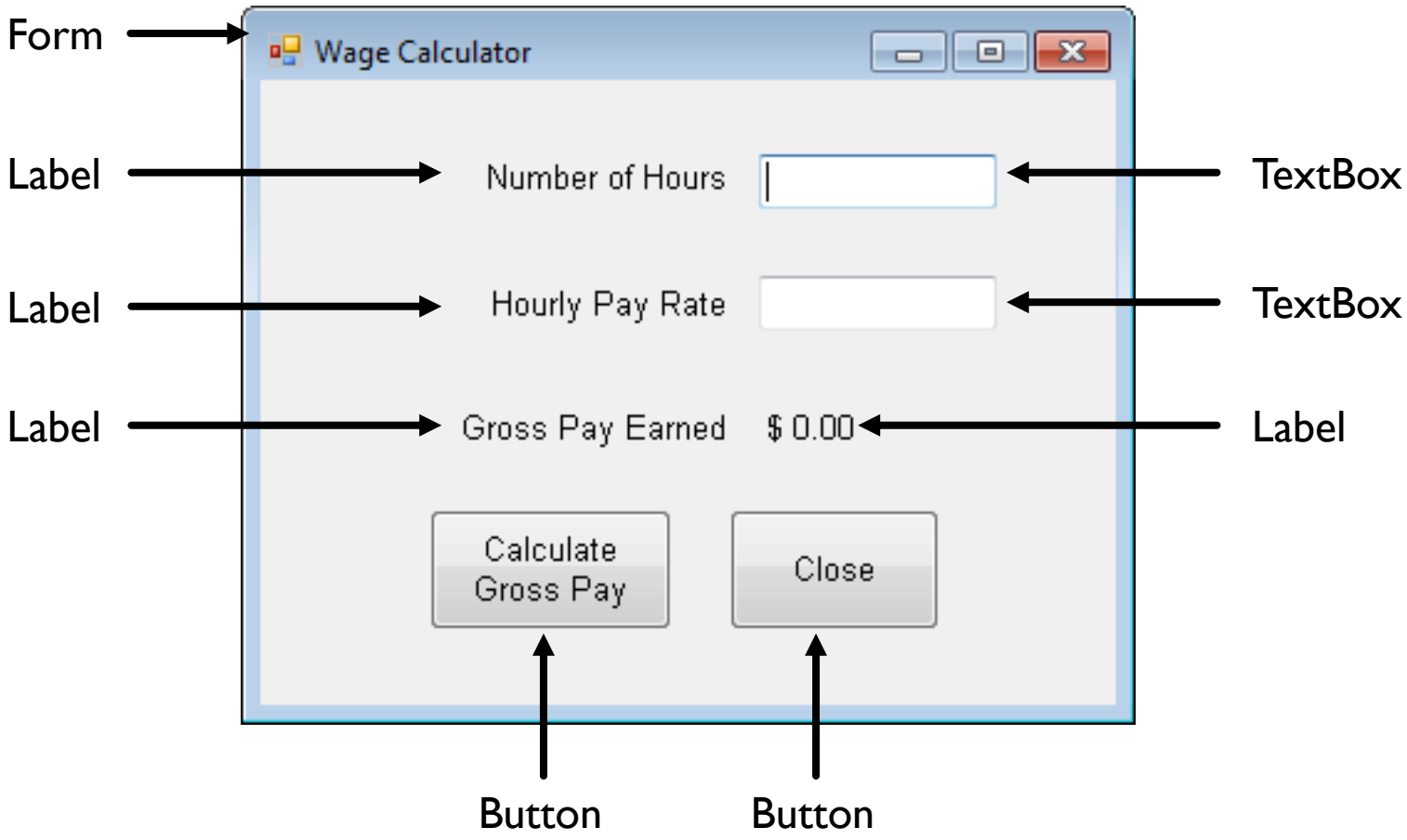
OBJECTS MODEL ANALOGY

- Class = automobile
- Properties of automobile class= make, model, color, engine, year
- Object = Each individual auto is an object.
  - Object is also an Instance of the automobile class.
- Methods = start, stop, speedup, slowdown
- Events of automobile class = Arrive, Crash

CONTROLS

- A **control** is specific type of object that usually appears in a program's graphical user interface
  - The window that contains the other elements is known as a Form control
  - The small boxes that accept input are known as TextBox controls
  - The areas that simply display text are known as Label controls
  - The buttons that perform operations when clicked with the mouse are known as Button controls

TYPES OF CONTROLS



VISUAL BASIC CONTROLS

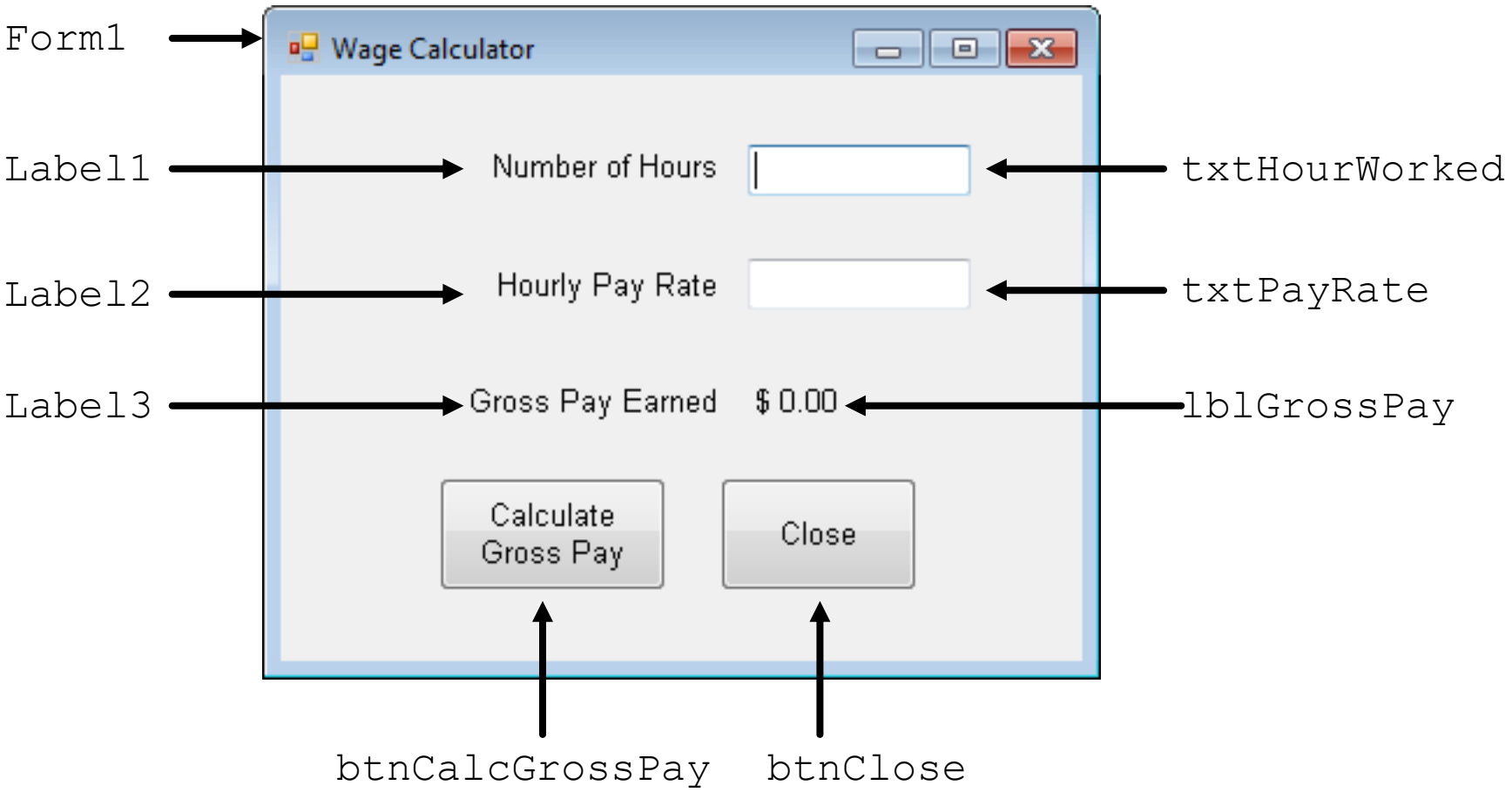
Table 1-2 Visual Basic controls

Control Type	Description
Button	A rectangular button-shaped object that performs an action when clicked with the mouse
CheckBox	A box that is checked or unchecked when clicked with the mouse
ComboBox	A control that is the combination of a ListBox and a TextBox
Form	A window, onto which other controls may be placed
GroupBox	A rectangular border that functions as a container for other controls
HScrollBar	A horizontal scroll bar that, when moved with the mouse, increases or decreases a value
Label	A box that displays text that cannot be changed or entered by the user
ListBox	A box containing a list of items
PictureBox	A control that displays a graphic image
RadioButton	A round button that is either selected or deselected when clicked with the mouse
TextBox	A rectangular area in which the user can enter text, or the program can display text
VScrollBar	A vertical scroll bar that, when moved with the mouse, increases or decreases a value

THE NAME PROPERTY

- All controls have properties
  - Each property has a value (or values)
  - Not all properties deal with appearance
- The Name property establishes a means for the program to refer to that control
  - Controls are assigned relatively meaningless names when created
  - Programmers usually change these names to something more meaningful

EXAMPLES OF CONTROL NAMES



CONTROL NAMING RULES AND CONVENTIONS

- Control names must start with a letter
- Remaining characters may be letters, digits, or underscore
- 1st 3 lowercase letters indicate the type of control
  - `txt...` for Text Boxes
  - `lbl...` for Labels
  - `btn...` for Buttons
- After that, capitalize the first letter of each word
  - `txtHoursWorked` is clearer than `txthoursworked`

WRITING VISUAL BASIC PROJECTS

- There is a three-step process when writing a Visual Basic application—you set up the user interface, define the properties, and then create the code.
- Planning
  - Design the User Interface.
  - Plan the Properties.
  - Plan the Basic Code; follow the language syntax rules; use pseudocode (English expression or comment describing action) then you move on to
- Programming (and use the same three-step process)
  - Define the User Interface.
  - Set the properties.
  - Write the Basic code.

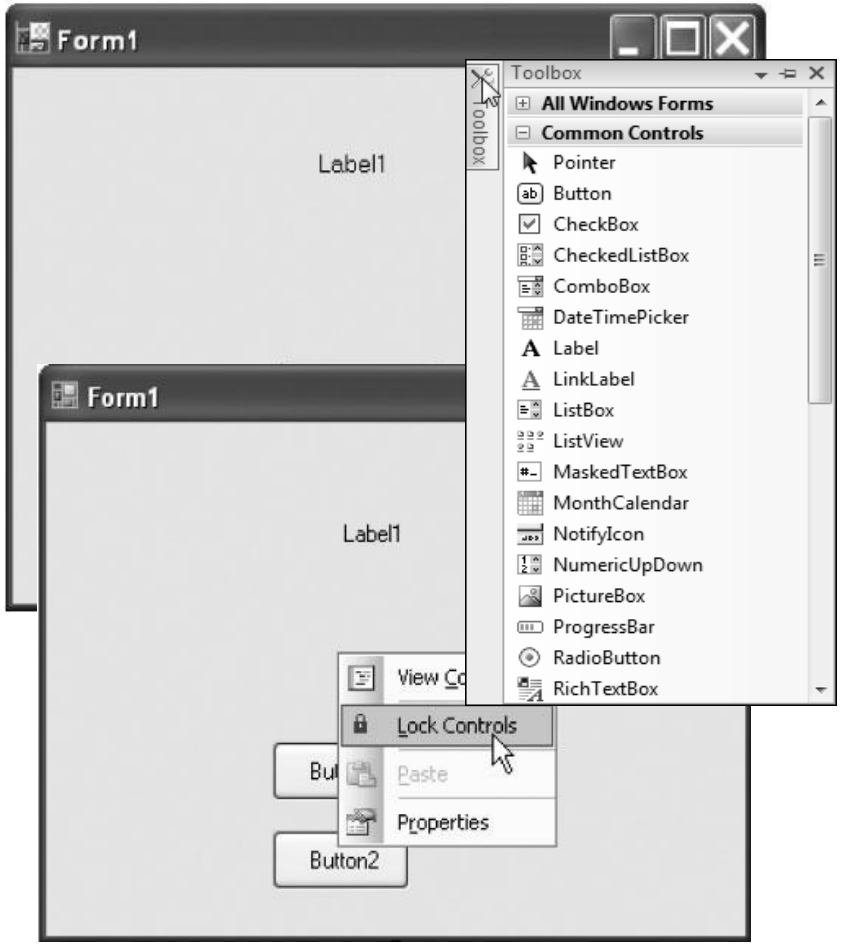
VBA IDE MODES

Design Time — used when designing the user interface and writing code
Run Time — used when testing and running a project
Break Time — if/when receiving a run-time error or pause error

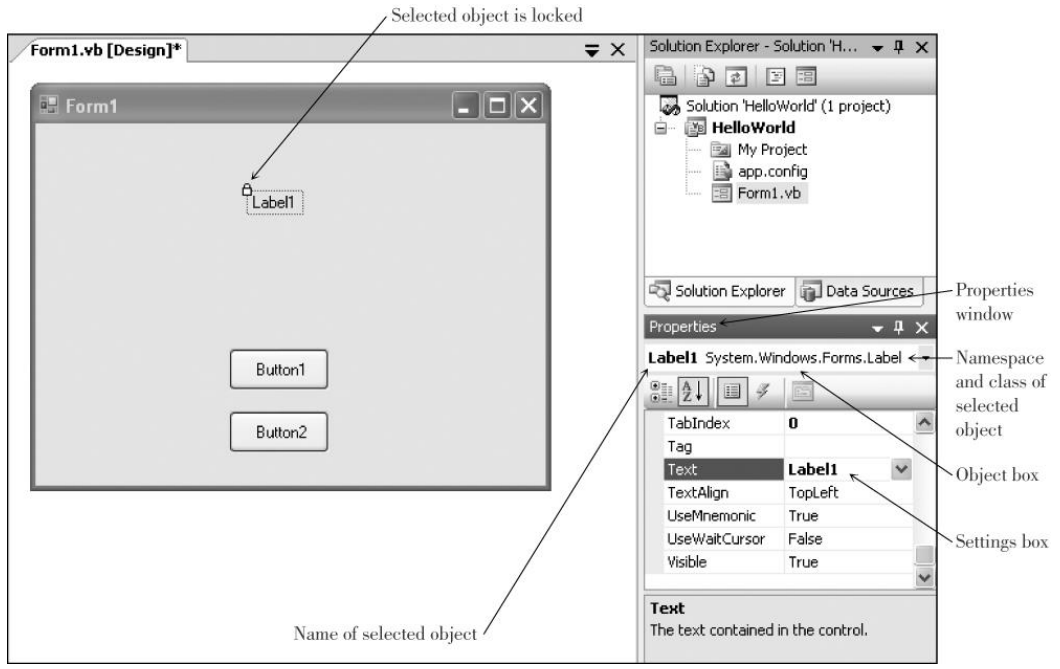
** The caption in the titlebar of the VBA IDE indicates which mode is currently active

PLANNING THE PROJECT

- Design the user interface.
  - Set up the form.
    - Resize the form.
    - Place a label and a button control on the form using the toolbox.
    - Lock the Controls in place.
- After the user interface is designed, the next step is to set the properties.

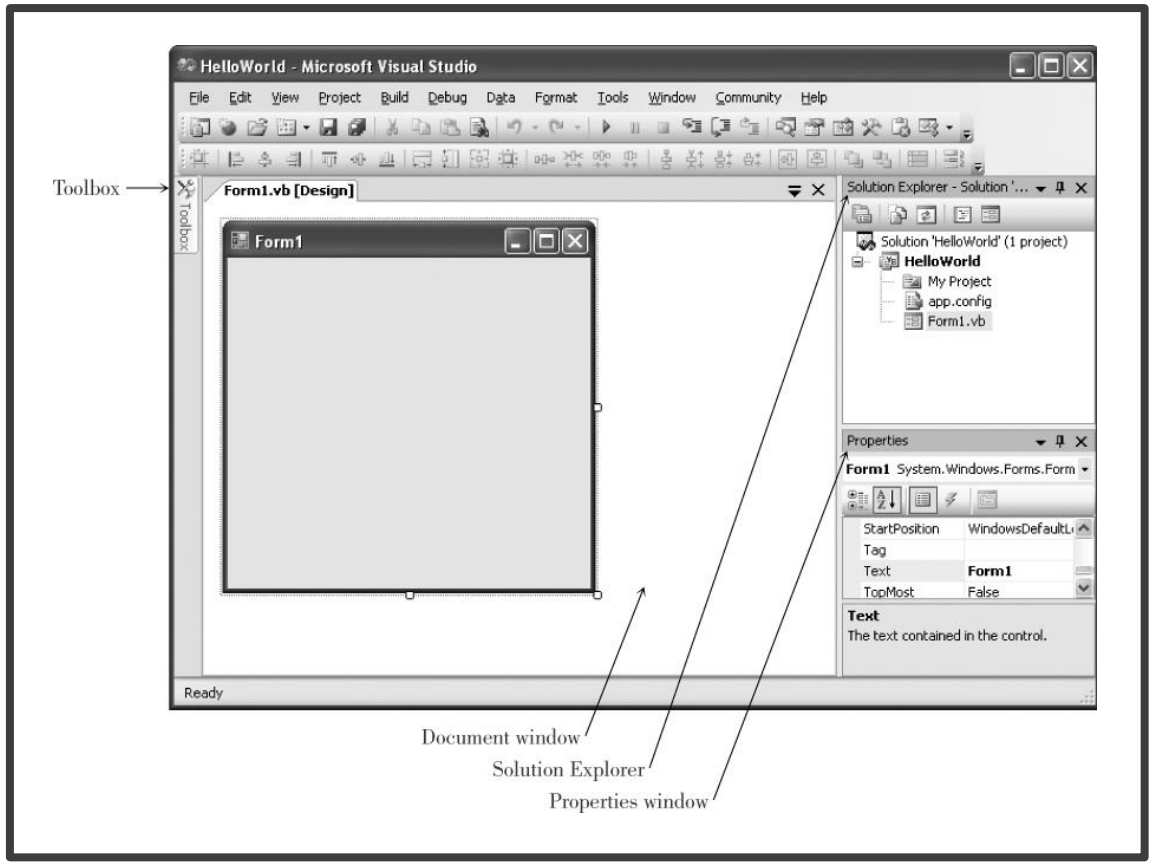


SETTING PROPERTIES



- **Label 1**
  - Name MessageLabel
  - Text leave blank
- **Button 1**
  - Name PushButton
  - Text Push Me
- **Button 2**
  - Name ExitButton
  - Text Exit
- **Form**
  - Name HelloForm
  - Text Hello World by *your name*

SETTING THE FORM PROPERTIES



- The default startup object is Form1
- The name of the form should always be changed to adhere to naming rules
- The properties window shows the files properties

WRITING THE CODE

- While the project is running, the user can perform actions.
- Each action by the user causes an event to occur.
- Write code for the events you care about; the events you want to respond to with code.
- Code is written as event procedures.
- VB will ignore events for which you do not write code.
- VB will automatically name event procedures as the object name, an underscore(_) and the name of the event.

MORE ON WRITING THE CODE

- When writing the code for your first project, you will use the following:
  - Remark Statement
  - Assignment Statement
  - Ending a Program
  - Editor Window

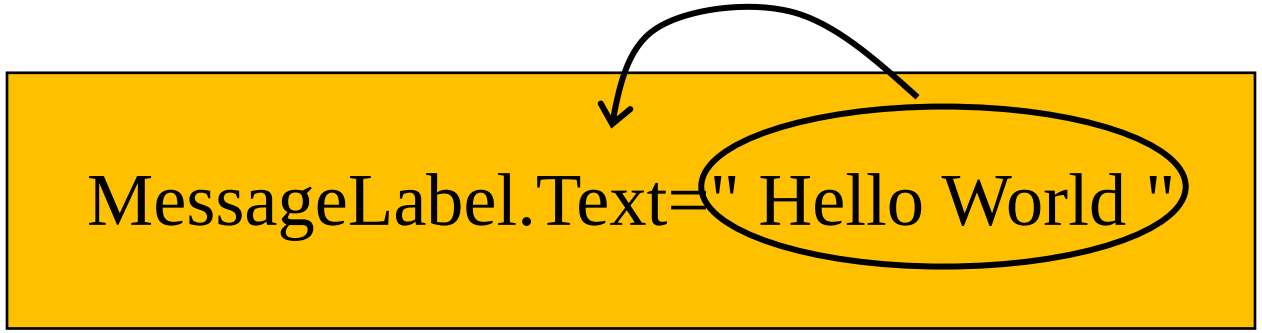
REMARK STATEMENT

- Also known as Comment, used for documentation; every procedure should begin with a remark statement providing explanation.
- Non-executable
- Automatically colored Green in Editor
- Begins with an apostrophe (')
  - On a separate line from executable code
  - At the right end of a line of executable code

```
'Display the Hello World message.
```

ASSIGNMENT STATEMENT

- Assigns a value to a property or variable
- Operates from right to left — the value appearing on the right side of the equal sign is assigned to the property named on the left of the equal sign.
- Enclose text strings in quotation marks (" ")



ENDING A PROGRAM

- Methods always have parentheses. (This will help you distinguish them from Properties which never have parentheses.)
- To execute a method of an object you write:

  Object.Method()
- Current Form may be referenced as **Me**

```
Me.Close( )
```

RUN, SAVE, MODIFY, PRINT, TEST, DEBUG, AND EXECUTE

- Run Project
  - Open Debug Menu, Start Debugging.
  - Start Debugging button on the toolbar.
  - Press F5, the *Start Debugging command.*
- Save Project — File Menu, Save All.
- Modify Project if needed.
- Print the Code.
- Correct any Errors and Rerun.
- When you start executing your program, the first step is called *compiling*, which means that the VB statements are converted to Microsoft Intermediate Language (MSIL). Your goal is to have no errors during the compile process: a **clean compile**.

"Help is always available from the Help Menu or by pressing F1."

FINDING AND FIXING ERRORS

- **Syntax Errors**
  - Breaks VB's rules for punctuation, format, or spelling
  - Smart editor finds most syntax errors, compiler finds the rest.
  - The editor identifies a syntax error with a squiggly blue line and you can point to an error to pop up the error message.
  - You can display the Error List window and line numbers in the source code to help locate the error lines.
- **Run-Time Errors**
  - Statements that fail to execute, such as impossible arithmetic operations
- **Logic Errors**
  - Project runs, but produces incorrect results.

END OF CHAPTER 1