

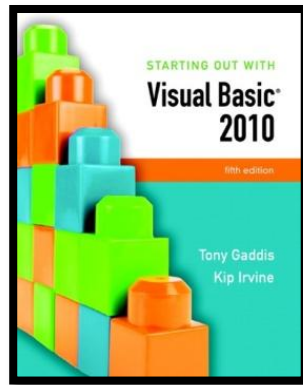


STARTING OUT WITH

Visual Basic® 2010

fifth edition

Tony Gaddis
Kip Irvine



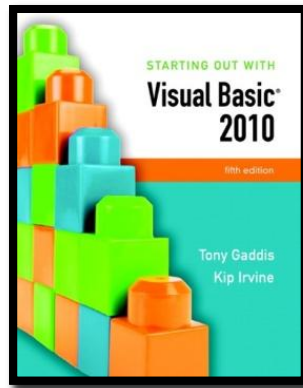
Chapter 1

Introduction to Programming and Visual Basic

Addison Wesley
is an imprint of



© 2011 Pearson Addison-Wesley. All rights reserved.



Section 1.1

COMPUTER SYSTEMS: HARDWARE AND SOFTWARE

Computer systems consist of similar hardware devices and hardware components. This section provides an overview of computer hardware and software organization.

Addison Wesley
is an imprint of

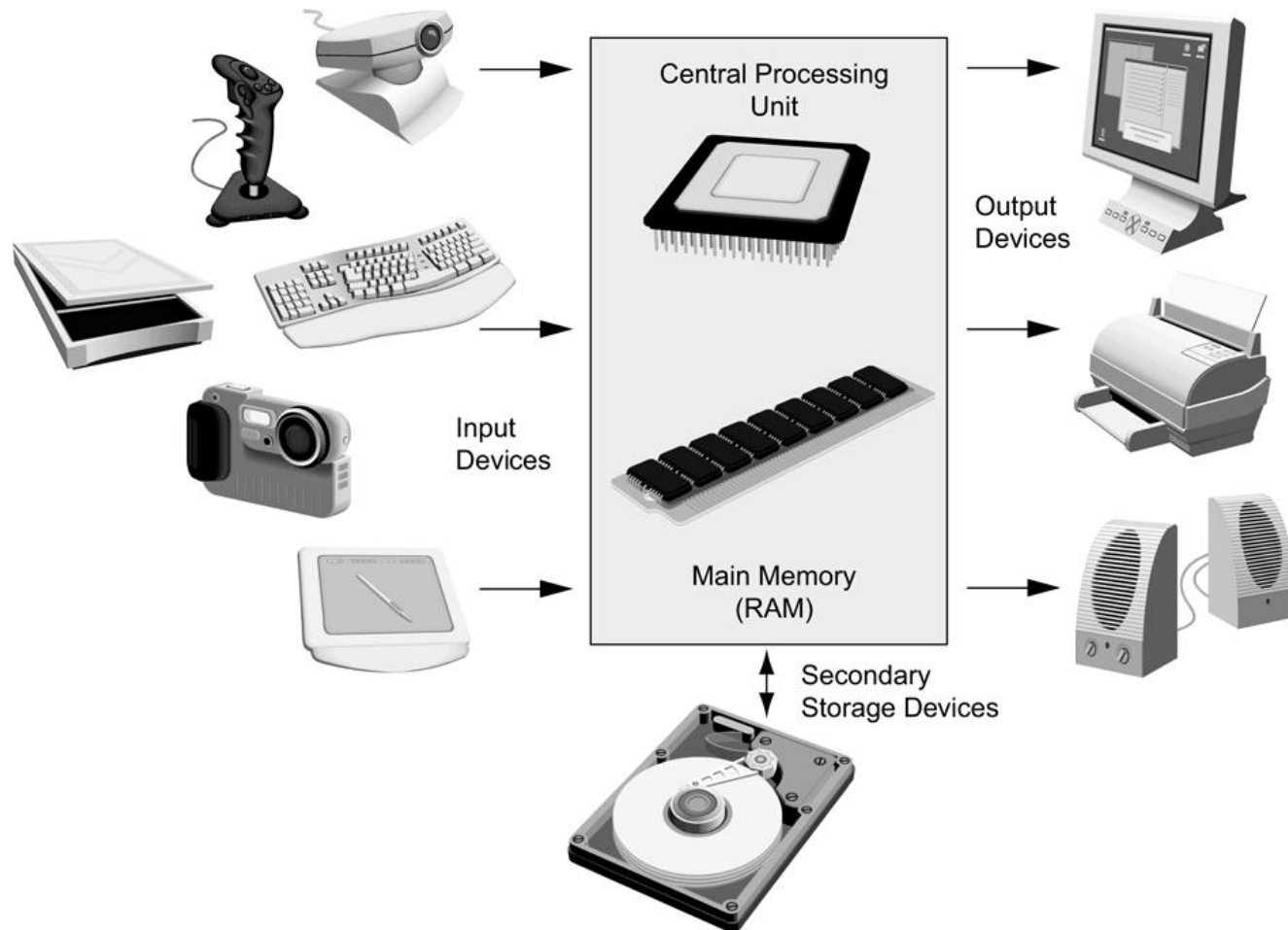


© 2011 Pearson Addison-Wesley. All rights reserved.

Computer Hardware

- Refers to the physical components
- Not one device but a system of many devices
- Major types of components include:
 - Central Processing Unit
 - Main memory
 - Secondary storage devices
 - Input devices
 - Output devices

Organization of a Computer System



The CPU

- Fetches instructions from main memory
- Carries out the operations commanded by the instructions
- Each instruction produces some outcome
- CPU gets instructions from a program
- A **program** is an entire sequence of instructions
- Instructions are stored as **binary numbers**
- **Binary number** - a sequence of 1's and 0's

Main Memory

- Commonly known as random access memory, or just RAM
- Holds instructions and data needed for programs that are currently running
- RAM is usually a **volatile** type of memory
- Contents are lost when power is turned off
- Used as temporary storage

Secondary Storage

- A nonvolatile storage medium
- Contents retained while power is off
- Hard disk drives are most common
- Records data magnetically on a circular disk
- Provides fast access to large amounts of data
- Optical devices store data on CD's as pits
- USB flash memory devices
- High capacity device plugs into USB port
- Portable, reliable, and fits easily in a pocket

Input Devices

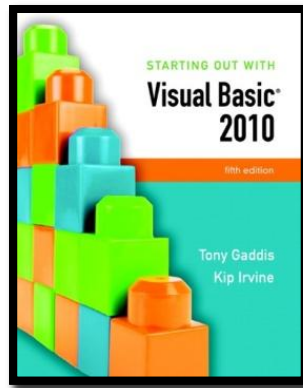
- Any type of device that provides data to a computer from the outside world
- For example:
 - Keyboard
 - Mouse
 - Scanner

Output Devices

- Any type of device that provides data from a computer to the outside world
- Examples of output data:
 - A printed report
 - An image such as a picture
 - A sound
- Common output devices include:
 - Monitor (display screen)
 - Printer

Software

- The programs that run on a computer
- Two major categories
 - Operating systems
 - Controls the processes within the computer
 - Manages the computer's hardware devices
 - Application Software
 - Solve problems or perform tasks needed by users
 - Examples include word processing, spreadsheets, games, Internet browsers, playing music, etc)
 - Each program is referred to as an application
 - This book develops applications in Visual Basic



Section 1.2

PROGRAMS AND PROGRAMMING LANGUAGES

A program is a set of instructions a computer follows in order to perform a task. A programming language is a special language used to write computer programs.

Addison Wesley
is an imprint of



© 2011 Pearson Addison-Wesley. All rights reserved.

What is a Program?

- Computers can only follow instructions
- A **computer program** is a set of instructions on how to solve a problem or perform a task
- In order for a computer to compute someone's gross pay, we must tell it to perform the steps on the following slide

Computing Gross Pay

1. Display message: "How many hours did you work?"
2. Allow user to enter number of hours worked
3. Store the number the user enters in memory
4. Display message: "How much are you paid per hour?"
5. Allow the user to enter an hourly pay rate
6. Store the number the user enters in memory
7. Multiply hours worked by pay rate and store the result in memory
8. Display a message with the result of the previous step

This well-defined, ordered set of steps for solving a problem is called an **algorithm**

States and Transitions

Memory snapshots show states of the program

3. Store hours worked in memory

6. Store hourly pay rate in memory

7. Multiply hours worked by pay rate and store amount earned in memory

Program Starting State

hours worked	??
--------------	----

hourly pay rate	??
-----------------	----

amount earned	??
---------------	----

Snapshot after Step 3

hours worked	20
--------------	----

hourly pay rate	??
-----------------	----

amount earned	??
---------------	----

Snapshot after Step 6

hours worked	20
--------------	----

hourly pay rate	25
-----------------	----

amount earned	??
---------------	----

Snapshot after Step 7

hours worked	20
--------------	----

hourly pay rate	25
-----------------	----

amount earned	500
---------------	-----

Programming Languages

- The steps in our algorithm must be stated in a form the computer understands
- The CPU processes instructions as a series of 1's and 0's called **machine language**
- This is a tedious and difficult format for people
- Instead, **programming languages** allow us to use words instead of numbers
- Software converts the programming language statements to machine language

Common Programming Languages

- Visual Basic
- Python
- JavaScript
- Java
- C#
- C
- C++
- PHP
- Visual Basic is not just a programming language
- It's a programming environment with tools to:
 - Create screen elements
 - Write programming language statements

What is a Program Made Of?

- **Keywords (Reserved Words)**
 - Words with special meaning that make up a high-level programming language, cannot be used for any other purpose
- **Operators**
 - Special symbols that perform various operations on data
- **Variables**
 - Used to store data in memory, named by the programmer
- **Syntax**
 - Set of rules
 - Similar to the syntax (rules) of a spoken language, such as English, but must be strictly followed
 - If even a single syntax error appears in a program, it will not compile or execute

What is a Program Made Of?

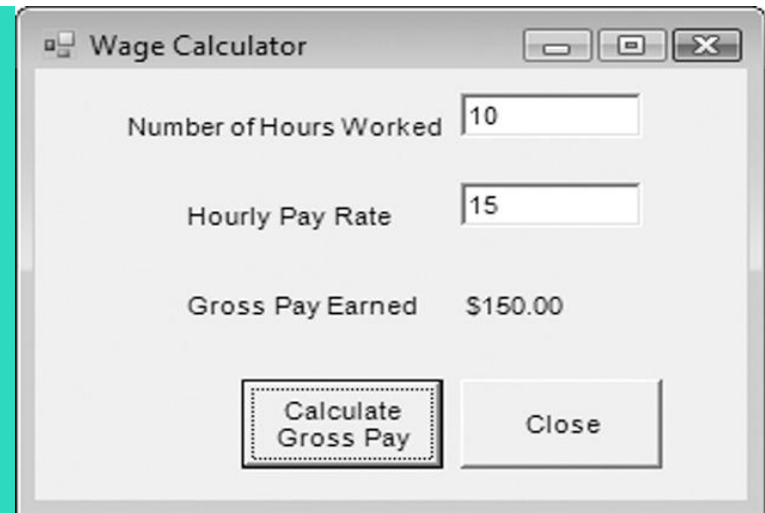
- Statements
 - Instructions made up of keywords, variables, and operators.
- Procedures
 - Set of programming statements that perform a specific task
- Comments (Remarks)
 - Ignored when the program runs, help human reader understand the purpose of programming statements
 - In Visual Basic, any statement that begins with an apostrophe (')

Methods of Programming

- Procedural
 - Constructed as a set of procedures (operational, functional units)
 - Each procedure is a set of instructions
 - The Gross Pay computation is a procedure
- Object-Oriented
 - Uses real-world **objects** such as students, transcripts, and courses
 - Objects have data elements called attributes
 - Objects also perform actions

Example of an Object

- This is a Visual Basic GUI object called a form
- Contains data and actions
- Data, such as Hourly Pay Rate, is a text **property** that determines the appearance of form objects
- Actions, such as Calculate Gross Pay, is a **method** that determines how the form reacts
- A form is an object that contains other objects such as buttons, text boxes, and labels



The screenshot shows a window titled "Wage Calculator" with a standard Windows-style title bar (minimize, maximize, close buttons). The form contains the following elements:

- A label "Number of Hours Worked" followed by a text box containing the value "10".
- A label "Hourly Pay Rate" followed by a text box containing the value "15".
- A label "Gross Pay Earned" followed by a text box containing the value "\$150.00".
- Two buttons at the bottom: "Calculate Gross Pay" (highlighted with a dashed border) and "Close".

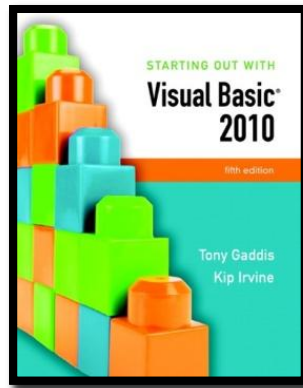
Example of an Object

- Form elements are objects called **controls**
- This form has:
 - Two **TextBox** controls
 - Four **Label** controls
 - Two **Button** controls
- The value displayed by a control is held in the **text** property of the control
- Left button text property is **Calculate Gross Pay**
- Buttons have methods attached to **click** events

The image shows a screenshot of a 'Wage Calculator' application window. The window contains the following elements: a label 'Number of Hours Worked' with a text box containing '10'; a label 'Hourly Pay Rate' with a text box containing '15'; a label 'Gross Pay Earned' with a text box containing '\$150.00'; a button labeled 'Calculate Gross Pay'; and a button labeled 'Close'. Arrows from the text on the left point to these elements: 'Number of Hours Worked' and 'Hourly Pay Rate' are identified as TextBox controls; 'Number of Hours Worked', 'Hourly Pay Rate', 'Gross Pay Earned', and 'Calculate Gross Pay' are identified as Label controls; and 'Calculate Gross Pay' and 'Close' are identified as Button controls.

Event Driven Programming: Events

- The GUI environment is event-driven
- An event is an action that takes place within a program
 - Clicking a button (a Click event)
 - Keying in a TextBox (a TextChanged event)
- Visual Basic controls are capable of detecting many, many events
- A program can respond to an event if the programmer writes an event procedure



Section 1.3

MORE ABOUT CONTROLS AND PROGRAMMING

As a Visual Basic programmer, you must design and create an application's GUI elements (forms and other controls) and the programming statements that respond to and/or perform actions (event handlers).

Addison Wesley
is an imprint of

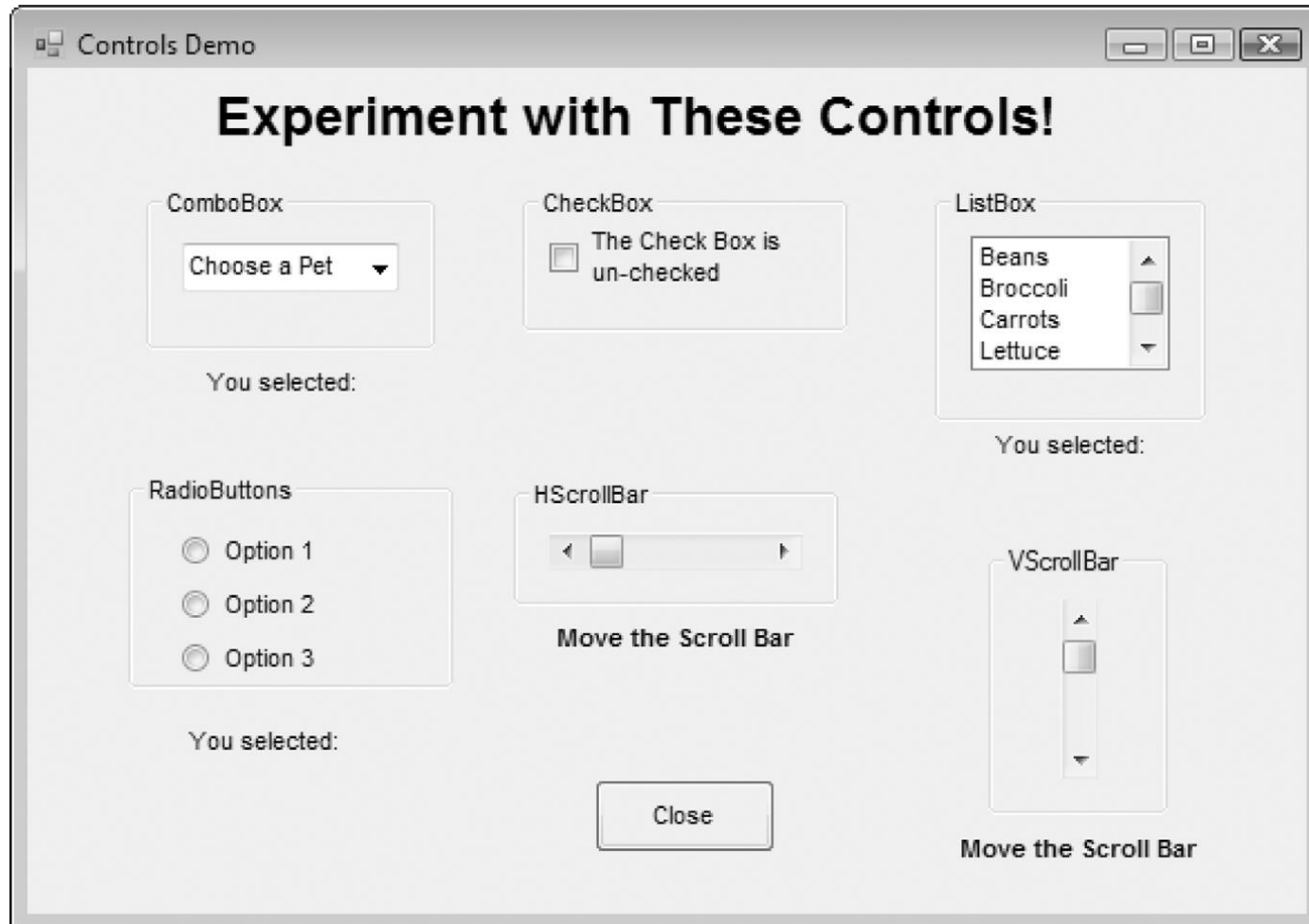


© 2011 Pearson Addison-Wesley. All rights reserved.

Visual Basic Controls

- As a Windows user you're already familiar with many Visual Basic controls:
 - Label - displays text the user cannot change
 - TextBox - allows the user to enter text
 - Button – performs an action when clicked
 - RadioButton - A round button that is selected or deselected with a mouse click
 - CheckBox – A box that is checked or unchecked with a mouse click
 - Form - A window that contains these controls
- Tutorial 1-3 demonstrates these controls

Tutorial 1-3, Visual Basic Controls

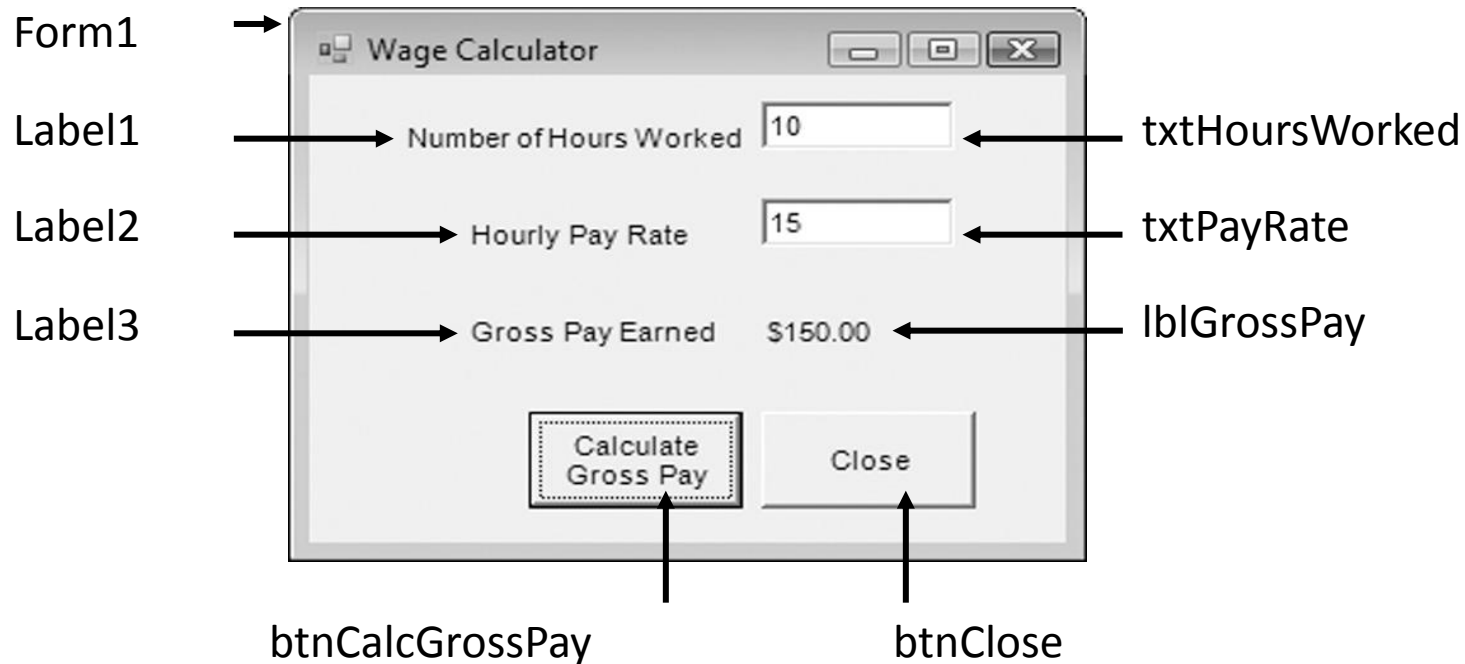


The Name Property

- 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

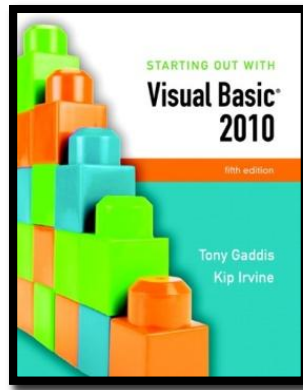
Examples of Names

- The Label controls use the default names (Label1, etc.)
- Text boxes, buttons, and the Gross Pay label play an active role in the program and have been changed



Naming 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



Section 1.4

THE PROGRAMMING PROCESS

The programming process consists of several steps, which include designing, creating, testing, and debugging activities.

Addison Wesley
is an imprint of



© 2011 Pearson Addison-Wesley. All rights reserved.

Step 1 of Developing an Application

- Clearly define what the program is to do
- For example, the Wage Calculator program:
 - Purpose: To calculate the user's gross pay
 - Input: Number of hours worked, hourly pay rate
 - Process: Multiply number of hours worked by hourly pay rate (result is the user's gross pay)
 - Output: Display a message indicating the user's gross pay

Step 2 of Developing an Application

- Visualize the application running on the computer and design its user interface

A hand-drawn user interface for a gross pay calculator. It features two input fields: "Number of Hours Worked" and "Hourly Pay Rate". Below these is a display field showing "Gross Pay Earned: \$0.00". At the bottom, there are two buttons: "Calculate Gross Pay" and "Close".

Number of Hours Worked

Hourly Pay Rate

Gross Pay Earned: \$0.00

Step 3 of Developing an Application

- Determine the controls needed

<u>Type</u>	<u>Name</u>	<u>Description</u>
TextBox	txtHoursWorked	Allows the user to enter the number of hours worked.
TextBox	txtPayRate	Allows the user to enter the hourly pay rate
Label	lblGrossPay	Displays the gross pay, after the btnCalcGrossPay button has been clicked
Button	btnCalcGrossPay	When clicked, multiplies the number of hours worked by the hourly pay rate
Button	btnClose	When clicked, terminates the application
Label	(default)	Description for Number of Hours Worked TextBox
Label	(default)	Description for Hourly Pay Rate TextBox
Label	(default)	Description for Gross Pay Earned Label
Form	(default)	A form to hold these controls

Step 4 of Developing an Application

- Define the values of each control's relevant properties

<u>Control Type</u>	<u>Control Name</u>	<u>Text</u>
Form	(Default)	"Wage Calculator"
Label	(Default)	"Number of Hours Worked"
Label	(Default)	"Hourly Pay Rate"
Label	(Default)	"Gross Pay Earned"
Label	lblGrossPay	"\$0.00"
TextBox	txtHoursWorked	""
TextBox	txtPayRate	""
Button	btnCalcGrossPay	"Calculate Gross Pay"
Button	btnClose	"Close"

Step 5 of Developing an Application

- Determine the event handlers and other code needed for each control

Method

btnCalcGrossPay_Click

Description

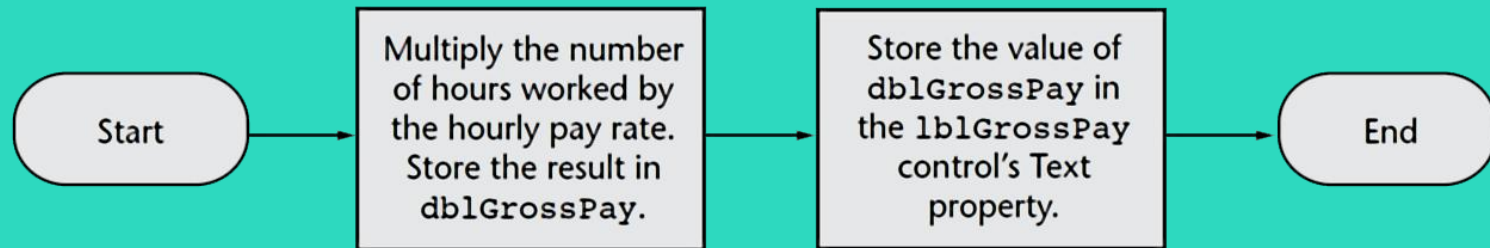
Multiplies hours worked by hourly pay rate
These values are entered into the
txtHoursWorked and txtPayRate TextBoxes
Result is stored in lblGrossPay Text property

btnClose_Click

Terminates the application

Step 6 of Developing an Application

- Create a flowchart or pseudocode version of the code
 - A **flowchart** is a diagram that graphically depicts the flow of a method



- **Pseudocode** is a cross between human language and a programming language

*Store Number of Hours Worked × Hourly Pay Rate in the grossPay variable.
Store the value of the grossPay variable in the lblGrossPay control's Text property.*

Step 7 of Developing an Application

- Check the code for errors:
 - Read the flowchart and/or pseudocode
 - Step through each operation as though **you** are the computer
 - Use a piece of paper to jot down the values of variables and properties as they change
 - Verify that the expected results are achieved

Step 8 of Developing an Application

- Use Visual Basic to create the forms and other controls identified in step 3
 - This is the first use of Visual Basic, all of the previous steps have just been on paper
 - In this step you develop the portion of the application the user will see

Step 9 of Developing an Application

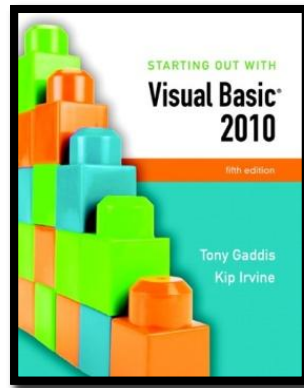
- Use the flowcharts or pseudocode from step 6 to write the actual code
 - This is the second step on the computer
 - In this step you develop the methods behind the click event for each button
 - Unlike the form developed on step 8, this portion of the application is invisible to the user

Step 10 of Developing an Application

- Attempt to run the application - find syntax errors
 - Correct any syntax errors found
 - **Syntax errors** are the incorrect use of an element of the programming language
 - Repeat this step as many times as needed
 - All syntax errors must be removed before Visual Basic will create a program that actually runs

Step 11 of Developing an Application

- Run the application using test data as input
 - Run the program with a variety of test data
 - Check the results to be sure that they are correct
 - Incorrect results are referred to as a runtime error
 - Correct any runtime errors found
 - Repeat this step as many times as necessary



Section 1.5

VISUAL STUDIO AND VISUAL BASIC EXPRESS EDITION (THE VISUAL BASIC ENVIRONMENT)

Visual Studio and Visual Basic Express Edition consist of tools that you use to build Visual Basic applications. The first step in using Visual Basic is learning about these tools.

Addison Wesley
is an imprint of



© 2011 Pearson Addison-Wesley. All rights reserved.

The Visual Studio IDE

- Visual Studio is an integrated development environment, often abbreviated as IDE
 - Provides everything needed to create, test, and debug software including:
 - The Visual Basic language
 - Form design tools to create the user interface
 - Debugging tools to help find and correct programming errors
- Visual Studio supports other languages beside Visual Basic such as C++ and C#

The Visual Basic Environment

- Tutorial 1-4 introduces elements of the IDE:
 - Customizing the IDE
 - Design window – a place to design and create a form
 - Solution Explorer window – shows files in the solution
 - Properties window – modify properties of an object
 - Dynamic Help window – a handy reference tool
 - Toolbar – contains icons for frequently used functions
 - Toolbox window – objects used in form design
 - Tooltips – a short description of button's purpose