Chapter 1
Introduction to Programming and Visual Basic 2005

1.1 Computer Systems: Hardware and Software

Computer Systems Consist of Similar Hardware Devices and Components

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
- 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 of RAM are lost when power is turned off

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

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
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

What is a program?

- 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 very tedious 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

- BASIC
- FORTRAN
- COBOL
- Pascal
- C
- C++
- C#
- Java

Visual Basic is not just a programming language
- It’s a programming environment with tools to:
  - Create screen elements
  - Write programming language statements

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
  - Represents real-world objects such as students, transcripts, and courses
  - Objects have data elements called attributes
  - Objects also perform actions called methods

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
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 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.

As a Visual Basic Programmer, You Must Design and Create the Two Major Components of an Application:
the GUI Elements (Forms and Other Controls) and the Programming Statements That Respond to And/or Perform Actions (Event Procedures).

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.

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

Event Handler – Compute Gross Pay
(GUI is on slide 19)
Private Sub btnCalcGrossPay_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCalcGrossPay.Click
    'Define a variable to hold the gross pay.
    Dim sngGrossPay As Single
    'Convert the values in the text boxes to numbers, and calculate the gross pay.
    sngGrossPay = CSng(txtHoursWorked.Text) * CSng(txtPayRate.Text)
    'Format the gross pay for currency display and assign it to the Text property of a label.
    lblGrossPay.Text = FormatCurrency(sngGrossPay)
End Sub

Event Handler - Close
Private Sub btnClose_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClose.Click
    'End the program by closing its window.
    Me.Close()
End Sub

Language Elements
- **Keywords**: Words with special meaning to Visual Basic (e.g., Private, Sub)
- **Programmer-defined-names**: Names created by the programmer (e.g., sngGrossPay, btnClose)
- **Operators**: Special symbols to perform common operations (e.g., +, -, *, and /)
- **Remarks**: Comments inserted by the programmer – these are ignored when the program runs (e.g., any text preceded by a single quote)

Language Elements: Syntax
- **Syntax** defines the correct use of key words, operators, & programmer-defined names
- Similar to the syntax (rules) of English that defines correct use of nouns, verbs, etc.
- A program that violates the rules of syntax will not run until corrected

The Programming Process
- **1.4 The Programming Process Consists of Several Steps, Which Include Design, Creation, Testing, and Debugging Activities**
Steps: Develop a VB Application
1. Clearly define what program is to do
2. Visualize application running on computer and design its user interface
3. Make list of controls needed
4. Define values of each control’s relevant properties
5. Make list of methods needed for each control
6. Create flowchart or pseudocode version for each method
7. Check the flowchart or pseudocode for errors
8. Start Visual Basic, create forms and other controls
9. Write code for event procedures and other methods
10. Attempt to run the application; correct syntax errors
11. Run application; use test data and verify results

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

Step 3 of Developing an Application
- Make a list of the controls needed

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TextBox</td>
<td>txtHoursWorked</td>
<td>Allows the user to enter the number of hours worked.</td>
</tr>
<tr>
<td>Label</td>
<td>lblHours</td>
<td>Description for Number of Hours Worked TextBox</td>
</tr>
<tr>
<td>Label</td>
<td>lblRate</td>
<td>Description for Hourly Pay Rate Label</td>
</tr>
<tr>
<td>Label</td>
<td>lblGross</td>
<td>Description for Gross Pay Earned Label</td>
</tr>
<tr>
<td>Label</td>
<td>lblGrossPay</td>
<td>Displays the gross pay, after the btnCalcGrossPay button has been clicked</td>
</tr>
<tr>
<td>Button</td>
<td>btnCalcGrossPay</td>
<td>Calculates gross pay when clicked, multiplies the number of hours worked by the hourly pay rate</td>
</tr>
<tr>
<td>Button</td>
<td>btnClose</td>
<td>Terminates the application when clicked</td>
</tr>
<tr>
<td>Label</td>
<td>lbStatus</td>
<td>Description for Gross Pay Earned Label</td>
</tr>
<tr>
<td>Form</td>
<td>frmCalcWage</td>
<td>A form to hold these controls</td>
</tr>
</tbody>
</table>

Step 4 of Developing an Application
- Define values for each control’s relevant properties:

<table>
<thead>
<tr>
<th>Control Type</th>
<th>Control Name</th>
<th>Text</th>
</tr>
</thead>
</table>
| Form         | frmCalcWage  | "Wage Calculator"
| Label        | lbStatus     | "Number of Hours Worked" |
| Label        | lbRate       | "Hourly Pay Rate" |
| Label        | lbGross      | "Gross Pay Earned" |
| Label        | lbGrossPay   | "$0.00" |
| TextBox      | txtHoursWorked | "Number of Hours WorkedTextBox" |
| TextBox      | txtPayRate   | "Hourly Pay RateTextBox" |
| Button       | btnCalcGrossPay | "Calculate Gross PayButton" |
| Button       | btnClose     | "CloseButton" |

Step 5 of Developing an Application
- List the methods needed for each control:

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>btnCalcGrossPay_Click</td>
<td>Multiplies hours worked by hourly pay rate These values are entered into the txtHoursWorked and txtPayRate TextBoxes Result is stored in lblGrossPay Text property</td>
</tr>
<tr>
<td>btnClose_Click</td>
<td>Terminates the application when clicked</td>
</tr>
</tbody>
</table>
Step 6 of Developing an Application

- Create pseudocode or a flowchart of each method:
  - Pseudocode is an English-like description in programming language terms
  - A flowchart is a diagram that uses boxes and other symbols to represent each step

  Store Hours Worked x Hourly Pay Rate in `sngGrossPay`. Store the value of `sngGrossPay` in `lblGrossPay.Text`.

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 Visual Basic to write the code for the event procedures and other methods created in step 6
  - 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
Visual Studio and the Visual Basic Environment

Visual Studio consists of tools that you use to build Visual Basic applications.

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