1
An Introduction to Computers and Visual Basic 2005

1.1 An Introduction to Computers  2

1.2 Using Windows  4
  ◆ Mouse Pointers ◆ Mouse Actions ◆ Windows Start Button
  ◆ Windows and Its Little Windows ◆ Using Notepad

1.3 Files and Folders  12
  ◆ Using Windows Explorer ◆ Using the Open and Save As Dialog Boxes ◆ Read-Only Attribute

1.4 An Introduction to Visual Basic 2005  18
  ◆ Why Windows and Why Visual Basic? ◆ How You Develop a Visual Basic Application ◆ The Different Versions of Visual Basic

1.5 Biographical History of Computing  21
  ◆ 1800s ◆ 1930s ◆ 1940s ◆ 1950s ◆ 1960s ◆ 1970s
  ◆ 1980s ◆ 1990s
Chapter 1  An Introduction to Computers and Visual Basic 2005

1.1  An Introduction to Computers

An Introduction to Programming Using Visual Basic 2005 is a book about problem solving using computers. The programming language used is Visual Basic 2005 (hereafter shortened to Visual Basic), but the principles taught apply to many modern programming languages. The examples and exercises present a sampling of the ways that computers are used in society.

Computers are so common today that you certainly have heard some of the terminology applied to them. Here are some questions that you might have about computers and programming.

**Question:** What is meant by a “personal” computer?

**Answer:** The word “personal” does not mean that the computer is intended for personal, as opposed to business, purposes. Rather, it indicates that the machine is operated by one person at a time instead of by many people.

**Question:** What are the main components of a personal computer?

**Answer:** Hidden from view inside the system unit are several components, including the microprocessor, memory, and hard drive of the computer. The central processing unit (CPU), sometimes referred to as the microprocessor, can be thought of as the computer’s brain, which carries out all of the computations. The memory, often referred to as random access memory (RAM), stores instructions and data while they are being used by the computer. When the computer’s power is turned off, the contents of memory are lost. A hard disk drive is used to store instructions and data when they are not being used in memory and when the computer is turned off. Inside the system unit there are also device cards, such as a graphics card, sound card, network card, and a modem. A graphics card is used to send an image to the monitor, and a sound card is used to send audio to a set of speakers attached to the computer. Network cards can be used to connect to a local area network (LAN) of computers, while a modem uses a telephone line to connect to any computer that can be reached by a phone call.

The personal computer also has several input and output devices, which are used to communicate with the computer. Standard input devices include the keyboard and mouse. Standard output devices include the monitor and printer. Instructions are entered into the computer by typing them on the keyboard, clicking a mouse, or loading them from a file located on a disk drive or downloaded from a network. Information processed by the computer can be displayed on the monitor, printed on the printer, or recorded on a disk drive.

**Question:** What are some uses of computers in our society?

**Answer:** The dramatic decrease in the cost of hardware and software technology has made computers widely available to consumers and corporations alike. Whenever we make a phone call, a computer determines how to route the call and calculates the cost of the call. Banks store all customer transactions on computers and process these transactions to revise the balance for each customer. Airlines record all reservations with computers. This information, which is stored in a database, can be accessed to determine the status of any flight. NASA uses computers to calculate the trajectories of satellites. Business analysts use computers to create pie and bar charts that give visual impact to data. With the Internet connecting millions of home computers, families and...
friends can exchange messages, information, and pictures. Consumers can shop from their PCs. Virtually no aspect of modern life is untouched by computer technology.

**Question:** What are some topics covered in this text that students could use immediately?

**Answer:** Computer files can be created to hold lists of names, addresses, and phone numbers, which can be alphabetized and displayed in their entirety or selectively. Mathematical computations can be carried out for science, business, and engineering courses. Personal financial transactions, such as bank deposits and loans, can be recorded, organized, and analyzed.

**Question:** How do we communicate with the computer?

**Answer:** Many languages are used to communicate with the computer. At the lowest level, there is machine language, which is understood directly by the microprocessor, but is awkward for humans. Visual Basic is an example of a higher-level language. It consists of instructions to which people can relate, such as Click, If, and Do.

**Question:** How do we get computers to perform complicated tasks?

**Answer:** Tasks are broken down into a sequence of instructions that can be expressed in a computer language. (This text uses the language Visual Basic.) This sequence of instructions is called a program. Programs can range in size from two or three instructions to millions of instructions. Instructions are typed on the keyboard, or read in from a file on a disk, and stored in the computer’s memory. The process of executing the instructions is called running the program.

**Question:** What is a server?

**Answer:** Whether a computer is a server depends on how it is being used. A computer that is only used by one person is a computer. A server is a computer that provides resources such as files, printers, or Internet access to other computers. Since a server needs to be continuously available to other computers, additional components are usually added to it to increase its reliability in the event of a power failure or other unexpected event.

**Question:** Are there certain features that all programs have in common?

**Answer:** Most programs do three things: take in data, manipulate them, and give desired information. These operations are referred to as input, processing, and output. The input data might be held in a portion of the program, reside on a disk drive, or be provided by the computer operator in response to requests made by the computer while the program is running. The processing of the input data occurs inside the computer and can take from a fraction of a second to many hours. The output data are either displayed on the monitor, printed on the printer, or recorded on a disk. As a simple example, consider a program that computes sales tax. An item of input data is the cost of the thing purchased. The processing consists of multiplying the cost by a certain percentage. An item of output data is the resulting product, the amount of sales tax to be paid.

**Question:** What are the meanings of the terms “hardware” and “software?”

**Answer:** Hardware refers to the physical components of the computer, including all peripherals, the central processing unit, disk drives, and all mechanical and electrical devices. Programs are referred to as software.
Question: What are the meanings of the terms “programmer” and “user?”

Answer: A programmer is a person who solves problems by writing programs on a computer. After analyzing the problem and developing a plan for solving it, he or she writes and tests the program that instructs the computer how to carry out the plan. The program might be run many times, either by the programmer or by others. A user is any person who uses a program. While working through this text, you will function both as a programmer and as a user.

Question: What is meant by problem solving?

Answer: Problems are solved by carefully reading them to determine what data are given and what outputs are requested. Then a step-by-step procedure is devised to process the given data and produce the requested output. This procedure is called an algorithm. Finally, a computer program is written to carry out the algorithm. Algorithms are discussed in Section 2.2.

Question: What types of problems are solved in this text?

Answer: Carrying out business computations, creating and maintaining records, alphabetizing lists, and displaying tabular data are some of the types of problems we will solve.

Question: How did Visual Basic 2005 evolve?

Answer: In the early 1960s, two mathematics professors at Dartmouth College developed BASIC to provide their students with an easily learned language that could tackle complicated programming projects. As the popularity of BASIC grew, refinements were introduced that permitted structured programming, which increases the reliability of programs. Visual Basic 1.0 is a version of BASIC developed in 1991 by the Microsoft Corporation to allow easy, visual-oriented development of Windows applications. Visual Basic 2005 is a language similar to the original Visual Basic, but more powerful. It is targeted for what is known as the .NET run time, which is a program that executes Visual Basic 2005 as well as programs from other languages that are targeted for the .NET run time. This will ultimately allow programs written in Visual Basic to be run on devices other than computers, such as cell phones and handheld devices. Other features of Visual Basic include full object-oriented programming capabilities and the development of Web services. Object-oriented programming is discussed in Chapter 11. The techniques presented in this book can be applied to the development of Web services.

1.2 Using Windows

Programs such as Visual Basic, which are designed for Microsoft Windows, are easy to use—once you learn a little jargon and a few basic techniques. This section explains the jargon, giving you enough understanding of Windows to get you started in Visual Basic. Although Windows may seem intimidating if you’ve never used it before, you need to learn only a few basic techniques, which are covered in this section.
1.2 Using Windows

Mouse Pointers

When you use Windows, think of yourself as the conductor and Windows as the orchestra. The conductor in an orchestra points to various members and does something with his or her baton; then the orchestra members respond in certain ways. For a Windows user, the baton is called the pointing device; most often it is a mouse. As you move the mouse across your desk, a pointer moves along the screen in sync with your movements. Two basic types of mouse pointers you will see in Windows are an arrow and an hourglass.

The arrow is the ordinary mouse pointer you use to point at various Windows objects before activating them. You will usually be told to “Move the pointer to . . .” This really means “Move the mouse around your desk until the mouse pointer is at . . .”

The hourglass mouse pointer pops up whenever Windows is saying “Wait a minute; I’m thinking.” This pointer still moves around when you move the mouse, but you can’t tell Windows to do anything until it finishes what it’s doing and the mouse pointer no longer resembles an hourglass. (Sometimes you can press the Esc key to tell Windows to stop what it is doing.)

Note: The mouse pointer can take on many other shapes, depending on which application you are using and what task you are performing. For instance, when entering text in a word processor or Visual Basic, the mouse pointer appears as a thin, large, uppercase I (referred to as an I-beam).

Mouse Actions

After you move the (arrow) pointer to a place where you want something to happen, you need to do something with the mouse. There are five basic things you can do with a mouse—point, hover, click, double-click, and drag.

Pointing means moving your mouse across your desk until the mouse pointer is over the desired object on the screen.

Hovering means lingering the mouse at a particular place and waiting for a message or menu to appear.

Clicking (sometimes people say single-clicking) means pressing and releasing the left mouse button once. Whenever a sentence begins “Click on . . .,” you need to

1. move the mouse pointer until it is at the object you are supposed to click on and
2. press and release the left mouse button.

An example of a sentence using this jargon might be “Click on the button marked Yes.” You also will see sentences that begin “Click inside the . . .” This means to move the mouse pointer until it is inside the boundaries of the object, and then click.

Double-clicking means clicking the left mouse button twice in quick succession (that is, pressing it, releasing it, pressing it, and releasing it again quickly so that Windows doesn’t think you single-clicked twice). Whenever a sentence begins “Double-click on . . .”, you need to

1. move the mouse pointer until it is at the object you are supposed to double-click on and
2. press and release the left mouse button twice in quick succession.
For example, you might be instructed to “Double-click on the little box at the far left side of your screen.”

Note: An important Windows convention is that clicking selects an object so you can give Windows or the document further directions about it, but double-clicking tells Windows to perform a default operation. For example, double-clicking on a folder will open that folder.

Dragging usually moves a Windows object. If you see a sentence that begins “Drag the . . .”, you need to

1. move the mouse pointer until it is at the object,
2. press the left mouse button and hold it down,
3. move the mouse pointer until the object moves to where you want it to be, and
4. finally, release the mouse button.

Sometimes this whole activity is called drag and drop.

- **Windows Start Button**

Clicking on the Start button at the bottom left corner of the screen displays a menu that you can use to run programs, shut down Windows, and carry out several other tasks. The Start menu also can be accessed by pressing a special key labeled with the Windows logo (located next to the Alt key) or by pressing Ctrl + Esc. (In the notation “key1 + key2”, the plus sign (+) instructs you to hold down key1 and then press key2. There are many useful key combinations of this type.)

- **Windows and Its Little Windows**

Windows gets its name from the way it organizes your screen into rectangular regions. When you run a program, the program runs inside a bordered rectangular box. Unfortunately Windows jargon calls all of these windows, so there’s only a lowercase “w” to distinguish them from the operating system called Windows.

When Windows’ attention is focused on a specific window, the Title bar at the top of the window is blue and the window is said to be active. (Inactive windows have a gray Title bar.) The active window is the only one that can be affected by your actions. An example of a sentence you might see is “Make the window active.” This means that if the Title bar of the window is gray, click inside the window. At this point, the active window will be responsive to your actions.

- **Using Notepad**

We will explore the Windows application Notepad to illustrate the Windows environment. Notepad is used extensively in this text to create text files for programs. Most of the concepts learned here carry over to Visual Basic and other Windows applications.

To invoke Notepad from Windows, click the Start button, click on Run, type “Notepad” into the box labeled “Name:”, and click the OK button. The window in Figure 1.1 will appear. As its name suggests, Notepad is an elementary word processor. You can type text into the Notepad window, edit the text, print the text on the printer, and save the text for later recall.
The blinking vertical line is called the **cursor**. Each letter you type will appear at the cursor. The Notepad window is divided into four parts. The part containing the cursor is called the **Work area**. It is the largest and most important part of the window because documents are typed into this window.

The **Title bar** at the top of the screen holds the name of the document currently being written. Until the document is given a name, the document is called "Untitled." The three buttons on the right side of the title bar can be used to maximize, minimize, or close the window. You can click on the **Maximize button** to make the Notepad window fill the entire screen, click on the **Minimize button** to change the Notepad window into a button on the taskbar, or click on the **Close button** to exit Notepad. As long as a window isn’t maximized or minimized, you can usually move it around the screen by dragging its title bar. (Recall that this means to move the mouse pointer until it is in the title bar, hold down the left mouse button, move the mouse until the window is where you want it to be, and then release the mouse button.)

**Note 1:** After you have maximized a window, the Maximize button changes to a pair of rectangles called the **Restore button**. Click on this button to return the window to its previous size.

**Note 2:** If the Notepad window has been minimized, it can be restored to its previous size by clicking on the button that was created on the taskbar when the application was opened. (The three tasks discussed in this paragraph also can be carried out with the **System Menu button** in the upper-left corner of the window.)

You can change the Notepad window to suit your needs. To adjust the size, do the following:

1. Move the mouse pointer until it is at the place on the boundary you want to adjust. The mouse pointer changes to a double-headed arrow.
2. Drag the border to the left or right or up or down to make the window smaller or larger.
3. When you are satisfied with the new size of the window, release the left mouse button.
If the Work area contains more information than can fit on the screen, you need a way to move through this information so you can see it all. For example, you will certainly be writing instructions in Visual Basic that are longer than one screen. You can use the mouse to scroll through your instructions with small steps or giant steps. A **Vertical scroll bar** lets you move from the top to the bottom of the window; a **Horizontal scroll bar** lets you move between the left and right margins of the window. Use this Scroll bar when the contents of the window are too wide to fit on the screen. Figure 1.1 shows both Vertical and Horizontal scroll bars.

A scroll bar has two arrows at the end of a channel and sometimes contains a box called the **Scroll box**. The Scroll box is the key to moving rapidly; the arrows are the key to moving in smaller increments. Dragging the Scroll box enables you to quickly move long distances to an approximate location in your document. For example, if you drag the Scroll box to the middle of the channel, you’ll scroll to approximately the middle of your document.

The **Menu bar** just below the Title bar is used to call up menus, or lists of tasks. Several of these tasks are described in this section.

Documents are created from the keyboard in much the same way they would be written with a typewriter. In computerese, writing a document is referred to as editing the document; therefore, Notepad is called a **text editor**.

After Notepad has been invoked, the following routine will introduce you to using Notepad:

1. Click on the Work area of Notepad.
2. Type a few words into Notepad.
3. Press the **Home key** to move the cursor back to the beginning of the line. In general, the Home key moves the cursor to the beginning of the line on which it currently is located.
4. Now press the **End** key. The cursor will move to the end of the line.
5. Type some letters, and then press the **Backspace** key a few times. It will erase letters one at a time. Another method of deleting a letter is to move the cursor to that letter and press the **Del** key. (Del stands for “Delete.”) The backspace key erases the character to the left of the cursor, and the Del key erases the character to the right of the cursor.
6. Hold down the **Ctrl** key (Ctrl stands for “Control”), and press the **Del** key. This key combination (denoted Ctrl + Del) erases the portion of the line to the right of the cursor.
7. Type more characters than can fit on one line of the screen. Notice that the leftmost characters scroll off the screen to make room for the new characters.
8. Press and release **Alt**, then press and release O, and then press and release W. (This key combination is abbreviated Alt/Format/Word Wrap or Alt/O/W. The slash character (/), officially called a **solidus**, instructs you to release the character preceding it, before pressing the character following it.) Notice that Notepad divided the long line so that it fits in Notepad’s window.
9. Click **Format** on the Menu bar, and notice that there is a check mark in front of Word Wrap. To remove the check mark, turn the Word Wrap feature off by clicking once on Word Wrap.
10. The **Enter** key is used to begin a new line on the screen.

11. The **Alt** key activates the Menu bar and causes a letter from each menu item to be underlined. Then, pressing one of the underlined letters, such as F, E, O, V, or H, selects a menu. (From the Menu bar, a menu also can be selected by pressing the right arrow key to highlight the name and then pressing the **Enter** key.) As shown in Figure 1.2, after a menu is opened, each option has one letter underlined. You can press an underlined letter to select an option. (Underlined letters are called **access keys**.) For instance, pressing A from the file menu selects the option “Save As”. Selections also can be made with the cursor-movement keys and the **Enter** key.

**Note 1:** You can select menus and options without the use of keys by clicking on them with the mouse.

**Note 2:** You can close a menu, without making a selection, by clicking anywhere outside the menu or pressing the **Esc** key twice.

12. The **Esc** key (Esc stands for “Escape”) is used to return to the Work area.

13. Press Alt/F/N. The dialog box in Figure 1.3 will appear and ask you if you want to save the current document. Decline by pressing N or clicking on the No button.

14. Type the following information into Notepad. (It gives the names of employees, their hourly wages, and the number of hours worked in the past week.) This document is used in Section 3.5

```
Mike Jones  
7.35 
35  
John Smith  
6.75 
33  
```
15. Store the document as a file on a disk. To save the document, press Alt/File/Save As. A dialog box appears, requesting a file name for the document. The cursor is in a narrow rectangular box labeled “File name:”. Type a drive letter, a colon, a backslash (\), and a name, and then press the Enter key or click on Save. For instance, you might type C:\PAYROLL. The document will then be stored on drive C. This process is called saving the document. Notepad automatically adds a period and the extension .txt to the name. Therefore, the complete file name is PAYROLL.TXT on the disk.

**Note:** If you want to save the document in a specific folder (directory) of the disk, include the folder’s name. For instance, you might type C:\Myfiles\PAYROLL. See Section 1.3 for a discussion of folders.

16. Press the key combination Alt/File/New to clear PAYROLL.TXT from Notepad.

17. To restore PAYROLL.TXT as the document in Notepad, press Alt/File/Open, type something like C:\PAYROLL in the “File name:” box, and then press the Enter key.

18. Move the cursor to the beginning of the document, and then press Alt/Edit/Find (or Ctrl + F) to invoke the Find dialog box. This dialog box contains several objects that will be discussed in this book. The text to be found should be typed into the rectangle containing the cursor. Such a rectangle is called a text box. The phrase “Find what:”, which identifies the type of information that should be placed into the text box, is referred to as the text of a label.

19. Type “smith” into the text box, and then click on the “Find Next” button. Clicking on it carries out a task. Text boxes, labels, and buttons are discussed in Section 3.2.

20. The small square to the left of the words “Match case” is called a check box. Click on it to see it checked, and then click again to remove the check mark.

21. The object captioned “Direction” is called a group box. It contains a pair of objects called radio buttons. Click on the “Up” radio button to select it, and then click on the “Down” radio button. Only one radio button at a time can be selected. Check boxes, group boxes, and radio buttons are discussed in Section 9.2.

22. Press Alt/File/Exit to exit Notepad.

**Comments**

1. Two useful key combinations that we have not discussed yet are the following:

   (a) Ctrl + Home moves the cursor to the beginning of the document.

   (b) Ctrl + End moves the cursor to the end of the document.
2. Notepad can perform many of the tasks of word processors, such as search and block operations. However, these features needn’t concern us presently. A discussion of them can be found in Appendix B, under “HOW TO: Use the Editor.”

Practice Problems 1.2

(Solutions to practice problems always follow the exercises.) Assume that you are using Windows Notepad:

1. Give two ways to open the Edit menu.
2. Assume that the Edit menu has been opened. Give three ways to pick a menu item.

EXERCISES 1.2

1. What does an hourglass pointer mean?
2. Describe “clicking” in your own words.
3. Describe “double-clicking” in your own words.
4. Describe “dragging” in your own words.
5. What is the blinking vertical line in Notepad called, and what is its purpose?
6. How can you tell when a window is active?
7. What is the difference between “Windows” and “windows?”
8. What is the purpose of the vertical scroll bar in Notepad?
9. By what name is a Notepad document known before it is named as part of being saved on disk?

In Exercises 10 through 24, give the key (or key combination) that performs the task in Windows Notepad.

10. Remove a pull-down menu from the screen.
11. Erase the character to the left of the cursor.
12. Access the Start menu.
13. Erase the character to the right of the cursor.
14. Move the cursor to the beginning of the line containing the cursor.
15. Move the cursor to the end of the line containing the cursor.
17. Move the cursor to the beginning of the document.
18. Move the cursor to the end of the document.
19. Move from the Work area to the Menu bar.
20. Move from the Menu bar to the Work area.
21. Cancel a dialog box.
22. Move from one option rectangle of a dialog box to another rectangle.
23. Save the current document on a disk.
24. Clear the current document from the Work area and start a new document.
Chapter 1 An Introduction to Computers and Visual Basic 2005

**Solutions to Practice Problems 1.2**

1. Press Alt>Edit, or click on the word Edit in the toolbar to display the Edit menu. The jargon says the menu is “dropped down” or “pulled down.”

2. Press the down-arrow key to highlight the item, then press the Enter key. Or, press the underlined letter in the name of the item, or click on the item.

### 1.3 Files and Folders

Modern computers have a hard disk, a diskette drive, and a CD (or DVD) drive. The hard disk is permanently housed inside the computer. You can read information from all three drives, but can write information easily only to the hard disk and to diskettes. We use the word disk to refer to either the hard disk, a diskette, a CD, or DVD. Each drive is identified by a letter. Normally, the hard drive is identified by C, the diskette drive by A, and the CD (or DVD) drive by D or E. Disk management is handled by Windows.

Disks hold not only programs, but also collections of data stored in files. The term file refers to either a program file, a text file, or some other kind of data file. We created a text file in Section 1.2. Each file has a name consisting of a base name followed by an optional extension consisting of a period and one or more characters. The term filename refers to the combination of the base name, the period, and the extension. A filename can contain up to 215 characters, typically consisting of letters, digits, spaces, periods, and other assorted characters. (The only characters that cannot be used in filenames are /, , *, ?, <, >, " , and |.) Extensions are normally used to identify the type of file. For example, spreadsheets created with Excel have the extension xls (eXceL Spreadsheet), documents created with Word have the extension doc (DOCument), and files created with Notepad have the extension txt (TeXT document). Some examples of filenames are “Annual Sales.xls,” “Letter to Mom.doc,” and “Phone.txt”.

Neither Windows nor Visual Basic distinguishes between uppercase and lowercase letters in folder and filenames. For instance, the names COSTS02.TXT, Costs02.Txt, and costs02.txt are equivalent. We use uppercase letters in this book for filenames.

Because a disk is capable of holding thousands of files, locating a specific file can be quite time consuming. Therefore, related files are grouped into collections called folders. For instance, one folder might hold all your Visual Basic programs, and another the documents created with your word processor.

Think of a disk as a large folder, called the root folder, that contains several smaller folders, each with its own name. (The naming of folders follows the same rules as the naming of files.) Each of these smaller folders can contain yet other named folders. Any folder contained inside another folder is said to be a subfolder of that folder. Each folder is identified by listing its name preceded by the names of the successively larger folders that contain it, with each folder name preceded by a backslash. Such a sequence is called a path. For instance, the path \Sales\NY02\July identifies the folder July, contained in the folder NY02, which in turn is contained in the folder Sales. Think of a file, along with its name, as written on a slip of paper that can be placed into either the root folder or one of the smaller folders. The combination of a drive letter followed by a colon, a path, and a filename is called a filespec, an abbreviation of “file specification.” Some examples of filespecs are C:\VB01\VB.EXE and A:\Personal\INCOME02.TXT.
In early operating systems such as MS-DOS, folders were called directories. Many Visual Basic objects and commands still refer to folders as directories. The terms “root folder” and “path” are a reference to the “tree” metaphor commonly used to describe a computer’s disk. In this metaphor, the large folder at the lowest level of the disk is called the “root” folder. The smaller folders contained in the root folder can be thought of as “branches” that emanate from the root. Each branch may have smaller branches, which in turn may have their own smaller branches, and so on. Finally, a file in one of these folders can be thought of as a leaf on a branch. The leaf is reached by starting at the root and following a “path” through the branches.

A program called Windows Explorer helps you view, organize, and manage the folders and files on your disks. We will learn how to use Windows Explorer to create, rename, copy, move, and delete folders and files.

**Using Windows Explorer**

To invoke Windows Explorer, click the Windows Start button, click on Run, type in the word “Explorer”, and click on the OK button. The appearance of the Explorer window depends on the version of Windows being used and the values of certain settings. Figure 1.4 (on the next page) shows a possible Explorer window for Windows XP. The Folders pane on the left side of the window contains a folder tree with the My Documents folder highlighted. (Only one folder at a time can be highlighted. The icon for a highlighted folder appears to be physically open, and its name appears in the title bar at the top of the Explorer window.) The contents of the highlighted folder are displayed in the right pane of the Explorer window. In Figure 1.4, the highlighted folder contains six subfolders and two files. To highlight a different folder, just click on it with the left mouse button.

In the Folders pane, you can click on a plus box to expand the folder tree so that it reveals the subfolders of the folder next to the plus box. You click on a minus box to reverse the process. This process allows you to locate any file.

In Figure 1.4, the folders and files in the right pane are displayed in the so-called Details view. This view is invoked by pressing Alt/V/D. Also, in Figure 1.4, the extensions of the filenames are shown. By default, Windows shows only the base names of files. The following steps get it to also display the extensions:

1. From Windows Explorer, press Alt/T/O to display the Folders Options dialog box.
2. Click on the View tab in the dialog box.
3. If there is a check mark in the box next to “Hide extensions for known file types,” click on the box to remove the check mark.
4. Click on the OK button to close the Folders Options dialog box.

To create a new folder:

1. Highlight the folder that is to contain the new folder as a subfolder.
2. On the File menu, point to New, and then click Folder. (Or press Alt/File/New/Folder.) The new folder appears with the temporary name New Folder.
3. Type a name for the folder, and then press the Enter key. (The allowable names for folders are the same as for files. However, folder names do not usually have an extension.)
Chapter 1  An Introduction to Computers and Visual Basic 2005

FIGURE 1.4  An Explorer window for Windows XP
To rename a folder or file:

1. Click on the folder or file in the right pane with the right mouse button.
2. In the Context menu that appears, click Rename. The current name will appear highlighted inside a rectangle.
3. Type the new name, and then press the Enter key.

To delete a folder or file:

1. Click on the folder or file with the right mouse button.
2. In the Context menu that appears, click Delete. A “Confirm Folder Delete” or a “Confirm File Delete” dialog box containing the name of the folder or file will appear.
3. Click the Yes button.

Or

1. Click on the file or folder with the left mouse button.
2. Press the Delete key.

To copy a folder or file:

1. Click on the folder or file to be copied with the right mouse button.
2. In the Context menu that appears, click on Copy.
3. Point to the folder where the copy is to be placed.
4. Click on the second folder with the right mouse button.
5. In the Context menu that appears, click on Paste.

To move a folder or file:

1. Click on the folder or file to be moved with the right mouse button.
2. In the Context menu that appears, click on Cut.
3. Point to the folder where the copy is to be moved.
4. Click on the second folder with the right mouse button.
5. In the Context menu that appears, click on Paste.

You also can carry out some of the preceding operations by “drag and drop.” For details, see the Help Topics accessed through the Windows Explorer Help menu. For instance, you can delete a folder or file by dragging it to the Recycle Bin and releasing the left mouse button.

**Using the Open and Save As Dialog Boxes**

In Section 1.2, we used the Open and Save As dialog boxes by just typing in the filespec for the desired file. These dialog boxes provide many features that assist with the locating of folders and files. Figure 1.5 shows an Open dialog box obtained from Windows XP. In the Save As dialog box, “Look in:” is replaced with “Save in:”, and “Files of type:” is replaced with “Save as type:”.

You can begin the search by clicking on one of the icons in the Places bar. In Figure 1.5, the My Computer icon was pressed. The following steps would be used to locate the file with filespec “C:\VB Programs\3-5-3\bin\Debug\PAYROLL.TXT”:

1. Double-click on “Local Disk (C:)” to obtain a list of all the folders and files on the hard drive C:. (The text in the “Look in:” box will now read “Local Disk (C:)”).
2. Double-click on the folder named “VB Programs” to obtain a list of its subfolders and files.
3. In succession, double-click on “3-5-3,” then on “bin,” and finally on “Debug.” The subfolders and files in the folder Debug will now be displayed. PAYROLL.TXT will be in the list.
4. Double-click on PAYROLL.TXT to open it.

The Save As dialog box operates in a similar way. However, after the desired folder is displayed in the “Save in:” box at the top of the dialog box, you would type the base name of the file into the “File name:” box.
Practice Problems 1.3

1. Where is the file having filespec “C:\TODAY.TXT” located?
2. Is “C:\Sales\New York” a filespec or a path?

EXERCISES 1.3

1. Explain why “Who is there?” is not a valid filename.
2. Explain why “FOUR STAR HOTEL ****” is not a valid filename.
3. What is wrong with the filespec “C:/Sports/TENNIS.DOC”?
4. Why do files on CDs usually have their read-only attribute turned on?
5. What is the path for a file whose filespec is “C:\Revenue\Chicago\MAIN.TXT”?
6. What is the filespec for the file PRES.TXT that is contained in the folder InfoUSA, where InfoUSA is a subfolder of the root directory of a diskette in the A drive?
7. Must the two files with filespecs “A:\DATA.TXT” and “A:\Info\DATA.TXT” be identical; that is, copies of one another?
8. What is the difference between a filespec and a filename?

From Windows Explorer, highlight a folder on your computer that contains many files, and then press Alt/V/D to select the Details option from the View menu. In Exercises 9-12, give the effect of clicking on the specified column head in the right-hand pane.

9. Size
10. Type
11. Modified
12. Name
13. The companion website for this book has a folder named Programs\Ch09\Pictures. Use Windows Explorer to obtain a list of the files in this folder, and then press Alt/V/H to select the Thumbnails option from the View menu. Describe what you see in the right pane.
14. Open the folder on your hard disk named My Documents. How many subfolders does the folder contain directly? How many files does the folder contain directly?

In Exercises 15 and 16, carry out the stated tasks.

15. (a) Take a blank diskette, and create two folders named Laurel and Hardy.
   (b) Create a subfolder of Laurel called Stan.
   (c) Use Notepad to create a file containing the sentence “Here’s another nice mess you’ve gotten me into.” and save the file with the name QUOTE.TXT in the folder Laurel.
   (d) Copy the file QUOTE.TXT into the folder Hardy.
   (e) Rename the new copy of the file QUOTE.TXT as LINE.TXT.
   (f) Delete the original copy of the file QUOTE.TXT.
16. (a) Take a blank diskette, create a folder named Slogans, and create two subfolders of Slogans named Coke and CocaCola.  
(b) Use Notepad to create a file containing the sentence “It's the real thing.”, and save the file with the name COKE1970.TXT in the folder Coke.  
(c) Use Notepad to create a file containing the phrase “The ideal brain tonic.”, and save the file with the name COKE1892.TXT in the folder Coke.  
(d) Copy the two files in Coke into the folder CocaCola.  
(e) Delete the folder Coke.  
(f) Rename the folder CocaCola as Coke.

Solutions to Practice Problems 1.3

1. The file is located in the root folder of the C drive.
2. It could be either. If “New York” is a folder, then it is a path. If “New York” is a file, it is a filespec. In this book, we always give extensions to files and never give extensions to folders. Therefore, by our conventions, “C:\Sales\New York” would be a path.

1.4 An Introduction to Visual Basic 2005

Visual Basic 2005 is one of the most exciting developments in programming in many years. It is the latest generation of Visual Basic, a language used by millions of software developers.

Visual Basic was designed to make user-friendly programs easier to develop. Prior to the creation of Visual Basic, developing a friendly user interface usually required a programmer to use a language such as C or C++, often requiring hundreds of lines of code just to get a window to appear on the screen. Now the same program can be created with much less time and fewer instructions using a language that is a direct descendant of BASIC—the language most accessible to beginning programmers.

Visual Basic requires the Microsoft Windows operating system. Although you don’t need to be an expert user of Microsoft Windows, you do need to know the basics before you can master Visual Basic—that is, you need to be comfortable with manipulating a mouse, you need to know how to manipulate a window, and you need to know how to use Notepad and Windows Explorer. However, there is no better way to master Microsoft Windows than to write applications for it—and that is what Visual Basic is all about.

Why Windows and Why Visual Basic?

What people call graphical user interfaces, or GUIs (pronounced “gooies”), have revolutionized the computer industry. Instead of the confusing prompts that earlier users once saw, today’s users are presented with a desktop filled with little pictures called icons. Icons provide a visual guide to what a program does or is used for.

Accompanying the revolution in how programs look was a revolution in how they feel. Consider a program that requests information for a database. Figure 1.6 shows how a program written before the advent of GUIs got its information. The program requests the six pieces of data one at a time, with no opportunity to go back and alter previously entered information. Then the screen clears and the six inputs are again requested one at a time. Figure 1.7 shows how an equivalent Visual Basic program gets its information.
Enter name (Enter EOD to terminate): Mr. President
Enter Address: 1600 Pennsylvania Avenue
Enter City: Washington
Enter State: DC
Enter Zipcode: 20500
Enter Phone Number: 202-456-1414

FIGURE 1.6 Input screen of a pre-VB program to fill a database.

The boxes may be filled in any order. When the user clicks on a box with the mouse, the cursor moves to that box. The user can either type in new information or edit the existing information. When the user is satisfied that all the information is correct, he or she just clicks on the Write to Database button. The boxes will clear, and the data for another person can be entered. After all names have been entered, the user clicks on the Exit button. In Figure 1.6, the program is in control; in Figure 1.7, the user is in control!

How You Develop a Visual Basic Application

One of the key elements of planning a Visual Basic application is deciding what the user sees—in other words, designing the screen. What data will he or she be entering? How large a window should the application use? Where will you place the buttons the user clicks on to activate the applications? Will the applications have places to enter text (text boxes) and places to display output? What kind of warning boxes (message boxes) should the application use? In Visual Basic, the responsive objects a program designer places on windows are called controls. Two features make Visual Basic different from traditional programming tools:

1. You literally draw the user interface, much like using a paint program.
2. Perhaps more important, when you’re done drawing the interface, the buttons, text boxes, and other objects that you have placed in a blank window will automatically recognize user actions such as mouse movements and button clicks.

FIGURE 1.7 Input screen of a Visual Basic program to fill a database.
That is, the sequence of procedures executed in your program is controlled by "events" that the user initiates rather than by a predetermined sequence of procedures in your program.

In any case, only after you design the interface does anything like traditional programming occur. Objects in Visual Basic recognize events like mouse clicks; how the objects respond to them depends on the instructions you write. You always need to write instructions in order to make controls respond to events. This makes Visual Basic programming fundamentally different from conventional programming.

Programs in traditional programming languages ran from the top down. For these programming languages, execution started from the first line and moved with the flow of the program to different parts as needed. A Visual Basic program works differently. Its core is a set of independent groups of instructions that are activated by the events they have been told to recognize. This event-driven methodology is a fundamental shift. The user decides the order in which things happen, not the programmer.

Most of the programming instructions in Visual Basic that tell your program how to respond to events like mouse clicks occur in what Visual Basic calls event procedures. Essentially, anything executable in a Visual Basic program either is in an event procedure or is used by an event procedure to help the procedure carry out its job. In fact, to stress that Visual Basic is fundamentally different from traditional programming languages, Microsoft uses the term project, rather than program, to refer to the combination of programming instructions and user interface that makes a Visual Basic application possible.

Here is a summary of the steps you take to design a Visual Basic application:

1. Design the appearance of the window that the user sees.
2. Determine the events that the controls on the window should recognize.
3. Write the event procedures for those events.

Now here is what happens when the program is running:

1. Visual Basic monitors the controls in the window to detect any event that a control can recognize (mouse movements, clicks, keystrokes, and so on).
2. When Visual Basic detects an event, it examines the program to see if you've written an event procedure for that event.
3. If you have written an event procedure, Visual Basic executes the instructions that make up that event procedure and goes back to Step 1.
4. If you have not written an event procedure, Visual Basic ignores the event and goes back to Step 1.

These steps cycle continuously until the application ends. Usually, an event must happen before Visual Basic will do anything. Event-driven programs are reactive more than active—and that makes them more user friendly.
1.5 Biographical History of Computing

The Different Versions of Visual Basic

Visual Basic 1.0 first appeared in 1991. It was followed by version 2.0 in 1992, version 3.0 in 1993, version 4.0 in 1995, version 5.0 in 1997, and version 6.0 in 1998. VB.NET, initially released in February 2002, was not backward compatible with the earlier versions of Visual Basic. It incorporated many features requested by software developers, such as true inheritance and powerful Web capabilities. Visual Basic 2005, released in November 2005, is a significantly improved version of VB.NET.

1.5 Biographical History of Computing

The following people made important contributions to the evolution of the computer and the principles of programming. While we think of the computer as a modern technology, it is interesting to note that many of its technologies and concepts were developed decades before Silicon Valley became an address in American culture.

1800s

George Boole: a self-taught British mathematician; devised an algebra of logic that later became a key tool in computer design. The logical operators presented in Section 5.1 are also known as Boolean operators.

Charles Babbage: a British mathematician and engineer; regarded as the father of the computer. Although the mechanical “analytical engine” that he conceived was never built, it influenced the design of modern computers. It had units for input, output, memory, arithmetic, logic, and control. Algorithms were intended to be communicated to the computer via punched cards, and numbers were to be stored on toothed wheels.

Augusta Ada Byron: a mathematician and colleague of Charles Babbage; regarded as the first computer programmer. She encouraged Babbage to modify the design based on programming considerations. Together they developed the concepts of decision structures, loops, and a library of procedures. Decision structures, loops, and procedures are presented in Chapters 5, 6, and 4 of this text, respectively.

Herman Hollerith: the founder of a company that was later to become IBM; at the age of 20, he devised a computer that made it possible to process the data for the U.S. Census of 1890 in one-third the time required for the 1880 census. His electromagnet- ic “tabulating machine” passed metal pins through holes in punched cards and into mercury-filled cups to complete an electric circuit. Each location of a hole corresponded to a characteristic of the population.

1930s

Alan Turing: a gifted and far-sighted British mathematician; made fundamental contributions to the theory of computer science, assisted in the construction of some of the early large computers, and proposed a test for detecting intelligence within a machine.
Chapter 1  An Introduction to Computers and Visual Basic 2005

His theoretical “Turing machine” laid the foundation for the development of general-purpose programmable computers. He changed the course of the Second World War by breaking the German “Enigma” code, thereby making secret German messages comprehensible to the Allies.

**John V. Atanasoff:** a mathematician and physicist at Iowa State University; declared by a federal court in Minnesota to be the inventor of the first electronic digital special-purpose computer. Designed with the assistance of his graduate assistant, Clifford Berry, this computer used vacuum tubes (instead of the less efficient relays) for storage and arithmetic functions.

**1940s**

**Howard Aiken:** a professor at Harvard University; built the Mark I, a large-scale digital computer functionally similar to the “analytical engine” proposed by Babbage. This computer took five years to build and used relays for storage and computations. It was technologically obsolete before it was completed.

**Grace M. Hopper:** retired in 1986 at the age of 79 as a rear admiral in the United States Navy; wrote the first major subroutine (a procedure that was used to calculate \( \sin x \) on the Mark I computer) and one of the first assembly languages. In 1945, she found that a moth fused onto a wire of the Mark I was causing the computer to malfunction, thus the origin of the term “debugging” for finding errors. As an administrator at Remington Rand in the 1950s, Dr. Hopper pioneered the development and use of COBOL, a programming language for the business community written in English-like notation.

**John Mauchley and J. Presper Eckert:** electrical engineers working at the University of Pennsylvania; built the first large-scale electronic digital general-purpose computer to be put into full operation. The ENIAC used 18,000 vacuum tubes for storage and arithmetic computations, weighed 30 tons, and occupied 1500 square feet. It could perform 300 multiplications of two 10-digit numbers per second, whereas the Mark I required 3 seconds to perform a single multiplication. Later they designed and developed the UNIVAC I, the first commercial electronic computer.

**John von Neumann:** a mathematical genius and member of the Institute for Advanced Study in Princeton, New Jersey; developed the stored program concept used in all modern computers. Prior to this development, instructions were programmed into computers by manually rewiring connections. Along with Hermann H. Goldstein, he wrote the first paper on the use of flowcharts.

**Maurice V. Wilkes:** an electrical engineer at Cambridge University in England and student of von Neumann; built the EDSAC, the first computer to use the stored program concept. Along with D. J. Wheeler and S. Gill, he wrote the first computer-programming text, *The Preparation of Programs for an Electronic Digital Computer* (Addison–Wesley, 1951), which dealt in depth with the use and construction of a versatile subroutine library.

**John Bardeen, Walter Brattain, and William Shockley:** physicists at Bell Labs; developed the transistor, a miniature device that replaced the vacuum tube and revolutionized computer design. It was smaller, lighter, more reliable, and cooler than the vacuum tube.
1950s

John Backus: a programmer for IBM; in 1953, headed a small group of programmers who wrote the most extensively used early interpretive computer system, the IBM 701 Speedcoding System. An interpreter translates a high-level language program into machine language one statement at a time as the program is executed. In 1957, Backus and his team produced the compiled language Fortran, which soon became the primary academic and scientific language. A compiler translates an entire program into efficient machine language before the program is executed. (Visual Basic combines the best of both worlds. It has the power and speed of a compiled language and the ease of use of an interpreted language.)

Reynold B. Johnson: IBM researcher; invented the computer disk drive. His disk drive, known as the Ramac, weighed a ton and stored five megabytes of data. Mr. Johnson’s other inventions included an electromechanical device that can read pencil-marked multiple-choice exams and grade them mechanically, the technology behind children’s “Talk to Me Books,” and major advances in the quality of tapes used in VCRs. He was a 1986 recipient of the National Medal of Technology.

Donald L. Shell: in 1959, the year that he received his Ph.D. in mathematics from the University of Cincinnati, published an efficient algorithm for ordering (or sorting) lists of data. Sorting often consumes a significant amount of running time on computers. The Shell sort is presented in Chapter 7 of this text.

1960s

John G. Kemeny and Thomas E. Kurtz: professors of mathematics at Dartmouth College and the inventors of BASIC; led Dartmouth to national leadership in the educational uses of computing. Kemeny’s distinguished career included serving as an assistant to both John von Neumann and Albert Einstein, serving as president of Dartmouth College, and chairing the commission to investigate the Three Mile Island nuclear power plant accident. In later years, Kemeny and Kurtz devoted considerable energy to the promotion of structured BASIC.

Corrado Bohm and Giuseppe Jacopini: European mathematicians; proved that any program can be written with the three structures discussed in Section 2.2: sequence, decisions, and loops. This result led to the systematic methods of modern program design known as structured programming.

Edsger W. Dijkstra: professor of computer science at the Technological University at Eindhoven, The Netherlands; stimulated the move to structured programming with the publication of a widely read article, “Go To Statement Considered Harmful.” In that article, he proposes that GOTO statements be abolished from all high-level languages such as BASIC. The modern programming structures available in Visual Basic do away with the need for GOTO statements.

Harlan B. Mills: IBM Fellow and professor of computer science at the University of Maryland; advocated the use of structured programming. In 1969, Mills was asked to write a program creating an information database for the New York Times, a project that was estimated to require 30 person-years with traditional programming techniques. Using structured programming techniques, Mills single-handedly completed...
the project in six months. The methods of structured programming are used throughout this text.

*Donald E. Knuth*: professor of computer science at Stanford University; generally regarded as the preeminent scholar of computer science in the world. He is best known for his monumental series of books, *The Art of Computer Programming*, the definitive work on algorithms.

*Ted Hoff, Stan Mazer, Robert Noyce, and Federico Faggin*: engineers at the Intel Corporation; developed the first microprocessor chip. Such chips, which serve as the central processing units for microcomputers, are responsible for the extraordinary reduction in the size of computers. A computer with greater power than the ENIAC now can be held in the palm of the hand.

*Douglas Engelbart*: human interface designer at the Stanford Research Institute; inventor of the computer mouse. While most of us would believe that the mouse is a new technology, the prototype was actually developed in the 1960s. Funded by a government project, Engelbert and his team developed the idea of a mouse to navigate a computer screen with pop-up “windows” to present information to the user. In a contest to choose the best navigation tool, the mouse won over the light pen, a joystick, a “nose-pointing” device, and even a knee-pointing device!

### 1970s

*Ted Codd*: software architect; laid the groundwork for relational databases in his seminal paper, “A Relational Model of Data for Large Shared Data Banks,” which appeared in the June 1970 issue of the *Communications of the ACM*. Relational databases are studied in Chapter 10 of this text.

*Paul Allen and Bill Gates*: cofounders of Microsoft Corporation; developed languages and the original operating system for the IBM PC. The operating system, known as MS-DOS, is a collection of programs that manage the operation of the computer. In 1974, Gates dropped out of Harvard after one year, and Allen left a programming job with Honeywell to write software together. Their initial project was a version of BASIC for the Altair, the first microcomputer. Microsoft is one of the most highly respected software companies in the World and a leader in the development of applications and programming languages.

*Stephen Wozniak and Stephen Jobs*: cofounders of Apple Computer Inc.; started the microcomputer revolution. The two had met as teenagers while working summers at Hewlett–Packard. Another summer, Jobs worked in an orchard, a job that inspired the names of their computers. Wozniak designed the Apple computer in Jobs’s parents’ garage, and Jobs promoted it so successfully that the company was worth hundreds of millions of dollars when it went public. Both men resigned from the company in 1985. Jobs founded a new company that developed the “Next” computer. He later returned to Apple and incorporated aspects of the Next computer into the Apple operating system.

*Dan Bricklin and Dan Fylstra*: cofounders of Software Arts; wrote VisiCalc, the first electronic spreadsheet program. An electronic spreadsheet is a worksheet divided into
rows and columns, which analysts use to construct budgets and estimate costs. A change made in one number results in the updating of all numbers derived from it. For instance, changing a person’s housing expenses will immediately produce a change in total expenses. Bricklin got the idea for an electronic spreadsheet after watching one of his professors at Harvard Business School struggle while updating a spreadsheet at the blackboard. VisiCalc became so popular that many people bought personal computers just so they could run the program.

**Dennis Ritchie:** member of the team at Bell Labs, creator of the C programming language. C is often referred to as a “portable assembly language.” Programs developed in C benefit from speed of execution by being fairly low-level and close to assembly language, yet not being tied up in the specifics of a particular hardware architecture. This characteristic was particularly important to the development of the Unix operating system, which occurred around the same time as the development of C. Throughout the 1970s, 1980s, 1990s, and even today, C has been a widely used language, particularly in situations where very fast program execution time is important.

**Ken Thompson:** member of the team at Bell Labs that created the Unix operating system as an alternative to the operating system for IBM’s 360 mainframe computers. Unlike many other earlier operating systems, Unix was written in C instead of assembly language. This allowed it to be adapted to a wide variety of computer architectures. Programmers could then develop programs in C that were intended to run on a Unix operating system, avoiding much of the rewriting involved in porting (adapting) these programs from one type of machine to another. Over the past 30 years, many variants upon Unix have emerged, often referred to as different “flavors” of Unix. Unix and its variants have played a tremendous role in the growth of the Internet, as well as being an operating system used by many commercial, scientific, and academic institutions.

**Alan Kay:** a brilliant programmer at the University of Utah; crystallized the concept of reusable building blocks of code to develop software programs. He developed a new language, Smalltalk, a pure object-oriented language, while at Xerox Palo Alto Research Center (PARC) in the 1970s. Most of today’s programming languages such as C++, C#, Java, and Visual Basic make use of object-oriented features first developed in Smalltalk. Still, because of its conceptual purity, Kay believes that Smalltalk “is the only real object-oriented language.”

**Don Chamberlain:** a Stanford Ph.D. and National Science Foundation scholar working at IBM; created a database programming language, later known as SQL (Structured Query Language). This innovative language was built on a “relational” model for data, where related data groups could be put into tables, then linked in various ways for easy programming and access. Very few people know that one of the world’s largest software companies, Oracle Corporation, was founded on this technology, developed by IBM and published for all to use. SQL is covered in Chapter 10 of this book.

### 1980s

**Phillip “Don” Estridge:** head of a product group at IBM; directly responsible for the success of the personal computer. The ubiquity of the PC today can be attributed to a marketing decision by Estridge to make off-the-shelf, easily producible computers for
a mass market, and to back that with IBM’s huge marketing resources. Estridge’s “skunk-works” group in Boca Raton broke many established IBM rules for product introduction. The IBM PC, introduced in 1981, chose an operating system from Microsoft and a processor chip from Intel over other vendors. This licensing deal opened the way for Microsoft’s and Intel’s successes today.

Mitchell D. Kapor: cofounder of Lotus Corporation; wrote the business software program Lotus 1-2-3, one of the most successful pieces of software for personal computers in its time. Lotus 1-2-3 is an integrated program consisting of a spreadsheet, a database manager, and a graphics package.

Tom Button: group product manager for applications programmability at Microsoft; headed the team that developed QuickBasic, QBasic, and Visual Basic. These modern, yet easy-to-use, languages greatly increased the productivity of programmers.

Alan Cooper: director of applications software for Coactive Computing Corporation; considered the father of Visual Basic. In 1987, he wrote a program called Ruby that delivered visual programming to the average user. A few years later, Ruby was combined with QuickBasic to produce Visual Basic, the remarkably successful language that allows Windows programs to be written from within Windows easily and efficiently.

Tim Berners–Lee: British computer scientist; father of the World Wide Web. He proposed the Web project in 1989 while working in Switzerland. His brainchild has grown into a global phenomenon. Berners-Lee, currently a senior research scientist at MIT, was awarded the first Millenium Technology Prize in 2004.

Charles Simonyi: a Hungarian programmer; known to the industry as the “father of Word.” He left his native Budapest as a 17-year-old prodigy to work at Xerox’s prestigious Palo Alto Research Center (PARC), where he developed the capability of “What You See Is What You Get” (WYSIWYG) software. This technology, which allows users to define the fonts and presentations for computer output, opened the door to desktop publishing on the personal computer. In 1980, Simonyi joined a fledgling software company called Microsoft and developed Microsoft Word into one of the most widely used software programs ever.

Bjarne Stroustrup: a native of Denmark; creator of the C++ programming language. Stoustrup came to the United States to work for Bell Labs, during which time he created C++ to extend the C programming language with additional capabilities for object-oriented and generic programming. C++ has been one of the most widely used programming languages, combining the speed and efficiency of C with features that make the development of large-scale programs much simpler. Because of its ability to work at a low level in a manner similar to C, C++ remains the language of choice for many projects where other programming languages such as Java and Visual Basic are not suitable.

Richard M. Stallman: a star programmer at MIT’s Artificial Intelligence Lab and a MacArthur Foundation Fellow; founded the Free Software Foundation (FSF). The FSF is an organization dedicated to promoting the free availability of software for public access, modification, and improvement. This philosophy contrasts with that of a large part of the commercial software development world, where software is developed for
sale, but the full rights to the source code are maintained by the company writing the software. Among his many technical accomplishments, Stallman created free versions of EMACS (a highly popular text editor on Linux/Unix systems) and GCC (a free C language compiler).

- **1990s**

**Marc Andreessen:** a former graduate student at the University of Illinois; inventor of the Web browser. He led a small band of fellow students to develop Mosaic, a program that allowed the user to move around the World Wide Web by clicking on words and symbols. Andreessen went on to cofound NCSA and Netscape Communications Corporation. Netscape’s was the leading Web browser throughout the mid 1990s before being surpassed by Microsoft’s Internet Explorer.

**James Gosling:** corporate vice president and Sun Fellow at Sun Microsystems; creator of the Java programming language. What started as an attempt to create a simple language for a networked world, Java, an object-oriented language, became a popular language for Internet programming. Java has become the primary teaching language at many universities. Visual Basic has many of the features and capabilities of Java.

**Linus Torvalds:** a graduate of the University of Helsinki in Finland; developed the popular Linux operating system. Linux began as a project by Linus to create a Unix operating system that could be used on personal computers. In the early 1990s, he began sharing the Linux source code with other OS programmers over the Internet, allowing them to contribute to and improve it. This philosophy resonated with the Internet culture, and the popularity of Linux grew quickly. Today, Linux is widely used, particularly as an operating system for Web servers. It is an open-source operating system, meaning that the source code (instructions) is made freely available for anyone to obtain, view, modify and use.