



Chapter 1

Introduction to Computers

and the Internet

Internet & World Wide Web
How to Program, 5/e



OBJECTIVES

In this chapter you'll learn:

- Computer hardware, soft-ware and Internet basics.
- The evolution of the Internet and the World Wide Web.
- How HTML5, CSS3 and JavaScript are improving web-application development.
- The data hierarchy.
- The different types of programming languages.
- Object-technology concepts.
- And you'll see demos of interesting and fun Internet applications you can build with the technologies you'll learn in this book.



- 1.1** Introduction
- 1.2** The Internet in Industry and Research
- 1.3** HTML5, CSS3, JavaScript, Canvas and jQuery
- 1.4** Demos
- 1.5** Evolution of the Internet and World Wide Web
- 1.6** Web Basics
- 1.7** Multitier Application Architecture
- 1.8** Client-Side Scripting versus Server-Side Scripting
- 1.9** World Wide Web Consortium (W3C)
- 1.10** Web 2.0: Going Social
- 1.11** Data Hierarchy



1.12 Operating Systems

1.12.1 Desktop and Notebook Operating Systems

1.12.2 Mobile Operating Systems

1.13 Types of Programming Languages

1.14 Object Technology

1.15 Keeping Up-to-Date with Information Technologies

1.1 Introduction

- ▶ The Internet and web programming technologies you'll learn in this book are designed to be *portable*, allowing you to design web pages and applications that run across an enormous range of Internet-enabled devices.
- ▶ *Client-side programming* technologies are used to build web pages and applications that are run on the *client* (i.e., in the browser on the user's device).
- ▶ *Server-side programming*—the applications that respond to requests from client-side web browsers, such as searching the Internet, checking your bank-account balance, ordering a book from Amazon, bidding on an eBay auction and ordering concert tickets.

1.1 Introduction (cont.)

- ▶ Read the Preface and the Before You Begin section to learn about the book's coverage and how to set up your computer to run the hundreds of code examples.
- ▶ The code is available at
www.deitel.com/books/iw3htp5 and
www.pearsonhighered.com/deitel.
- ▶ Use the source code to run every program and script as you study it.
- ▶ Try each example in *multiple browsers*.

1.1 Introduction (cont.)

- ▶ If you're interested in smartphones and tablet computers, run the examples in your browsers on iPhones, iPads, Android smartphones and tablets, and others.
- ▶ The technologies covered in this book and browser support for them are evolving rapidly. *Not every feature of every page we build will render properly in every browser.*

1.1 Introduction (cont.)

Moore's Law

- ▶ Every year or two, the capacities of computers have approximately *doubled* inexpensively.
- ▶ This remarkable trend often is called **Moore's Law**.
- ▶ Moore's Law and related observations apply especially to the amount of memory that computers have for programs, the amount of secondary storage (such as disk storage) they have to hold programs and data over longer periods of time, and their processor speeds—the speeds at which computers execute their programs (i.e., do their work).
- ▶ Similar growth has occurred in the communications field, in which costs have plummeted as enormous demand for communications bandwidth (i.e., information-carrying capacity) has attracted intense competition.



1.2 The Internet in Industry and Research

- ▶ Figures 1.1–1.4 provide a few examples of how computers and the Internet are being used in industry and research.
- ▶ Figure 1.1 lists two examples of how computers and the Internet are being used to improve health care.



Name	Description
Electronic health records	These might include a patient's medical history, prescriptions, immunizations, lab results, allergies, insurance information and more. Making this information available to health care providers across a secure network improves patient care, reduces the probability of error and increases overall efficiency of the health care system.
Human Genome Project	The Human Genome Project was founded to identify and analyze the 20,000+ genes in human DNA. The project used computer programs to analyze complex genetic data, determine the sequences of the billions of chemical base pairs that make up human DNA and store the information in databases which have been made available over the Internet to researchers in many fields.

Fig. 1.1 | Computers and the Internet in health care.

1.2 The Internet in Industry and Research (cont.)

- ▶ Figure 1.2 provides a sample of some of the exciting ways in which computers and the Internet are being used for social good.



Name	Description
AMBER™ Alert	The AMBER (America's Missing: Broadcast Emergency Response) Alert System is used to find abducted children. Law enforcement notifies TV and radio broadcasters and state transportation officials, who then broadcast alerts on TV, radio, computerized highway signs, the Internet and wireless devices. AMBER Alert recently partnered with Facebook, whose users can "Like" AMBER Alert pages by location to receive alerts in their news feeds.
World Community Grid	People worldwide can donate their unused computer processing power by installing a free secure software program that allows the World Community Grid (www.worldcommunitygrid.org) to harness unused capacity. This computing power, accessed over the Internet, is used in place of expensive supercomputers to conduct scientific research projects that are making a difference, providing clean water to third-world countries, fighting cancer, growing more nutritious rice for regions fighting hunger and more.

Fig. 1.2 | Projects that use computers and the Internet for social good.



Name	Description
One Laptop Per Child (OLPC)	One Laptop Per Child (one.laptop.org) is providing low-power, inexpensive, Internet-enabled laptops to poor children worldwide—enabling learning and reducing the digital divide.

Fig. 1.2 | Projects that use computers and the Internet for social good.

1.2 The Internet in Industry and Research (cont.)

- ▶ Figure 1.3 gives some examples of how computers and the Internet provide the infrastructure to communicate, navigate, collaborate and more.



Name	Description
Cloud computing	Cloud computing allows you to use software, hardware and information stored in the “cloud”—i.e., accessed on remote computers via the Internet and available on demand—rather than having it stored on your personal computer. Amazon is one of the leading providers of public cloud computing services. You can rent extra storage capacity using the Amazon Simple Storage Service (Amazon S3), or augment processing capabilities with Amazon’s EC2 (Amazon Elastic Compute Cloud). These services, allowing you to increase or decrease resources to meet your needs at any given time, are generally more cost effective than purchasing expensive hardware to ensure that you have enough storage and processing power to meet your needs at their peak levels. Business applications (such as CRM software) are often expensive, require significant hardware to run them and knowledgeable support staff to ensure that they’re running properly and securely. Using cloud computing services shifts the burden of managing these applications from the business to the service provider, saving businesses money.

Fig. 1.3 | Examples of computers and the Internet in infrastructure. (Part 1 of 3.)



Name	Description
GPS	Global Positioning System (GPS) devices use a network of satellites to retrieve location-based information. Multiple satellites send time-stamped signals to the GPS device, which calculates the distance to each satellite based on the time the signal left the satellite and the time the signal arrived. This information is used to determine the exact location of the device. GPS devices can provide step-by-step directions and help you easily find nearby businesses (restaurants, gas stations, etc.) and points of interest. GPS is used in numerous location-based Internet services such as check-in apps to help you find your friends (e.g., Foursquare and Facebook), exercise apps such as RunKeeper that track the time, distance and average speed of your outdoor jog, dating apps that help you find a match nearby and apps that dynamically update changing traffic conditions.
Robots	Robots can be used for day-to-day tasks (e.g., iRobot's Roomba vacuum), entertainment (e.g., robotic pets), military combat, deep sea and space exploration (e.g., NASA's Mars rover) and more. RoboEarth (www.roboearth.org) is "a World Wide Web for robots." It allows robots to learn from each other by sharing information and thus improving their abilities to perform tasks, navigate, recognize objects and more.

Fig. 1.3 | Examples of computers and the Internet in infrastructure. (Part 2 of 3.)

Name	Description
E-mail, Instant Messaging, Video Chat and FTP	Internet-based servers support all of your online messaging. E-mail messages go through a mail server that also stores the messages. Instant messaging (IM) and Video Chat apps, such as AIM, Skype, Yahoo! Messenger and others allow you to communicate with others in real time by sending your messages and live video through servers. FTP (file transfer protocol) allows you to exchange files between multiple computers (e.g., a client computer such as your desktop and a file server) over the Internet using the TCP/IP protocols for transferring data.

Fig. 1.3 | Examples of computers and the Internet in infrastructure. (Part 3 of 3.)

1.2 The Internet in Industry and Research (cont.)

- ▶ Figure 1.4 lists a few of the exciting ways in which computers and the Internet are used in entertainment.



Name	Description
iTunes and the App Store	iTunes is Apple's media store where you can buy and download digital music, movies, television shows, e-books, ringtones and apps (for iPhone, iPod and iPad) over the Internet. Apple's iCloud service allows you to store your media purchases "in the cloud" and access them from any iOS (Apple's mobile operating system) device. In June 2011, Apple announced at their World Wide Developer Conference (WWDC) that 15 billion songs had been downloaded through iTunes, making Apple the leading music retailer. As of July 2011, 15 billion apps had been downloaded from the App Store (www.apple.com/pr/library/2011/07/07Apples-App-Store-Downloads-Top-15-Billion.html).
Internet TV	Internet TV set-top boxes (such as Apple TV and Google TV) allow you to access an enormous amount of content on demand, such as games, news, movies, television shows and more.

Fig. 1.4 | Examples of computers and the Internet in entertainment. (Part 1 of 2.)



Name	Description
Game programming	Global video game revenues are expected to reach \$65 billion in 2011 (uk.reuters.com/article/2011/06/06/us-videogames-factbox-idUKTRE75552I20110606). The most sophisticated games can cost as much as \$100 million to develop. Activision's <i>Call of Duty 2: Modern Warfare</i> , released in 2009, earned \$310 million in just one day in North America and the U.K. (news.cnet.com/8301-13772_3-10396593-52.html?tag=mncol;txt)! Online <i>social gaming</i> , which enables users worldwide to compete with one another over the Internet, is growing rapidly. Zynga—creator of popular online games such as <i>Farmville</i> and <i>Mafia Wars</i> —was founded in 2007 and already has over 265 million monthly users. To accommodate the growth in traffic, Zynga is adding nearly 1,000 servers each week (techcrunch.com/2010/09/22/zynga-moves-1-petabyte-of-data-daily-adds-1000-servers-a-week/)!

Fig. 1.4 | Examples of computers and the Internet in entertainment. (Part 2 of 2.)



1.3 HTML5, CSS3, JavaScript, Canvas and jQuery

HTML5

- ▶ HTML (HyperText Markup Language) is a special type of computer language called a *markup language* designed to specify the *content* and *structure* of web pages (also called documents) in a portable manner.
- ▶ HTML5, now under development, is the emerging version of HTML.
- ▶ HTML enables you to create content that will render appropriately across the extraordinary range of devices connected to the Internet—including smartphones, tablet computers, notebook computers, desktop computers, special-purpose devices such as large-screen displays at concert arenas and sports stadiums, and more.



1.3 HTML5, CSS3, JavaScript, Canvas and (cont.)

- ▶ A “stricter” version of HTML called *XHTML* (*Extensible HyperText Markup Language*), which is based on XML (eXtensible Markup Language), is still used frequently today.
- ▶ Many of the server-side technologies we cover later in the book produce web pages as XHTML documents, by default, but the trend is clearly to HTML5.



1.3 HTML5, CSS3, JavaScript, Canvas and (cont.)

Cascading Style Sheets (CSS)

- ▶ Although HTML5 provides some capabilities for controlling a document's presentation, *it's better not to mix presentation with content*.
- ▶ **Cascading Style Sheets (CSS)** are used to specify the *presentation*, or styling, of elements on a web page (e.g., fonts, spacing, sizes, colors, positioning).
- ▶ CSS was designed to style portable web pages *independently* of their content and structure.
- ▶ By separating page styling from page content and structure, you can easily change the look and feel of the pages on an *entire* website, or a portion of a website, simply by swapping out one style sheet for another.
- ▶ CSS3 is the current version of CSS under development.



1.3 HTML5, CSS3, JavaScript, Canvas and (cont.)

JavaScript

- ▶ JavaScript helps you build *dynamic* web pages (i.e., pages that can be modified “on the fly” in response to *events*, such as user input, time changes and more) and computer applications.
- ▶ It enables you to do the client-side programming of web applications.
- ▶ JavaScript was created by Netscape.
- ▶ Both Netscape and Microsoft have been instrumental in the standardization of JavaScript by ECMA International (formerly the European Computer Manufacturers Association) as ECMAScript.
- ▶ ECMAScript 5, the latest version of the standard, corresponds to the version of JavaScript we use in this book.
- ▶ JavaScript is a portable scripting language. Programs written in JavaScript can run in web browsers across a wide range of devices.



1.3 HTML5, CSS3, JavaScript, Canvas and (cont.)

Web Browsers and Web-Browser Portability

- ▶ Ensuring a consistent look and feel on client-side browsers is one of the great challenges of developing web-based applications.
- ▶ Currently, a standard does not exist to which software vendors must adhere when creating web browsers.
- ▶ Although browsers share a common set of features, each browser might render pages differently.



1.3 HTML5, CSS3, JavaScript, Canvas and (cont.)

- ▶ Browsers are available in many versions and on many different platforms (Microsoft Windows, Apple Macintosh, Linux, UNIX, etc.).
- ▶ Vendors add features to each new version that sometimes result in cross-platform incompatibility issues.
- ▶ It's difficult to develop web pages that render correctly on all versions of each browser.
- ▶ All of the code examples in the book were tested in the five most popular desktop browsers and the two most popular mobile browsers (Fig. 1.5).

Browser	Approximate market share as of August 2011 (http:// gs.statcounter.com)	Score out of 450 from html5test.com
Desktop browsers	Market share	
Google Chrome 13	17%	330
Mozilla Firefox 6	27%	298
Apple Safari 5.1	7%	293
Opera 11.5	2%	286
Internet Explorer 9	40%	141
Mobile browsers	Mobile market share	
iPhone	15% (of mobile browsers)	217
Android	18% (of mobile browsers)	184

Fig. 1.5 | HTML5 Test scores for the browsers used to test the examples.



Portability Tip 1.1

The web is populated with many different browsers, including many older, less-capable versions, which makes it difficult for authors and web-application developers to create universal solutions. The W3C is working toward the goal of a universal client-side platform (<http://www.w3.org/2006/webapi/admin/charter>).



1.3 HTML5, CSS3, JavaScript, Canvas and (cont.)

- ▶ Support for HTML5, CSS3 and JavaScript features varies by browser.
- ▶ The *HTML5 Test* website (<http://html5test.com/>) scores each browser based on its support for the latest features of these evolving standards.
- ▶ Figure 1.5 lists the five desktop browsers we use in reverse order of their HTML5 Test scores from most compliant to least compliant at the time of this writing.
- ▶ You can also check sites such as <http://caniuse.com/> for a list of features covered by each browser.



1.3 HTML5, CSS3, JavaScript, Canvas and (cont.)

jQuery

- ▶ jQuery (jQuery.org) is currently the most popular of hundreds of *JavaScript libraries*.
 - www.activoinc.com/blog/2008/11/03/jquery-emerges-as-most-popular-javascript-library-for-web-development/.
- ▶ jQuery simplifies JavaScript programming by making it easier to manipulate a web page's elements and interact with servers in a portable manner across various web browsers.
- ▶ It provides a library of custom graphical user interface (GUI) controls (beyond the basic GUI controls provided by HTML5) that can be used to enhance the look and feel of your web pages.



1.3 HTML5, CSS3, JavaScript, Canvas and (cont.)

Validating Your HTML5, CSS3 and JavaScript Code

- ▶ You must use proper HTML5, CSS3 and JavaScript syntax to ensure that browsers process your documents properly.
- ▶ Figure 1.6 lists the validators we used to validate the code in this book. Where possible, we eliminated validation errors.

Technology	Validator URL
HTML5	http://validator.w3.org/ http://html5.validator.nu/
CSS3	http://jigsaw.w3.org/css-validator/
JavaScript	http://www.javascriptlint.com/ http://www.jslint.com/

Fig. 1.6 | HTML5, CSS3 and JavaScript validators.

1.4 Demos

- ▶ Browse the web pages in Fig. 1.7 to get a sense of some of the things you'll be able to create using the technologies you'll learn in this book, including HTML5, CSS3, JavaScript, canvas and jQuery.



URL	Description
https://developer.mozilla.org/en-US/demos/	Mozilla's DemoStudio contains numerous HTML5, canvas, CSS3 and JavaScript demos that use audio, video, animation and more.
http://js-fireworks.appspot.com/	Enter your name or message, and this JavaScript animation then writes it using a fireworks effect over the London skyline.
http://9elements.com/io/projects/html5/canvas/	Uses HTML5 canvas and audio elements to create interesting effects, and ties in tweets that include the words “HTML5” and “love” (click anywhere on the screen to see the next tweet).
http://www.zachstronaut.com/lab/text-shadow-box/text-shadow-box.html	Animated demo of the CSS3 text-shadow effect. Use the mouse to shine a light on the text and dynamically change the direction and size of the shadow.

Fig. 1.7 | HTML5, CSS3, JavaScript, canvas and jQuery demos.
(Part 1 of 4.)



URL	Description
http://clublime.com/lab/html5/sphere/	Uses an HTML5 canvas to create a sphere that rotates and changes direction as you move the mouse cursor.
http://spielzeugz.de/html5/liquid-particles.html	The Liquid Particles demo uses an HTML5 canvas. Move the mouse around the screen and the “particles” (dots or letters) follow.
http://www.paulbrunt.co.uk/bert/	Bert’s Breakdown is a fun video game built using an HTML5 canvas.
http://www.openrise.com/lab/Flower-Power/	Canvas app that allows you to draw flowers on the page, adjust their colors, change the shapes of the petals and more.
http://alteredqualia.com/canvasmol/	Uses canvas to display a 3D molecule that can be viewed from any desired angle (0–360 degrees).
http://pasjans-online.pl/	The game of Solitaire built using HTML5.

Fig. 1.7 | HTML5, CSS3, JavaScript, canvas and jQuery demos.
(Part 2 of 4.)



URL	Description
http://andrew-hoyer.com/experiments/cloth/	Uses canvas to simulate of the movement of a piece of cloth. Click and drag the mouse to move the fabric.
http://www.paulrhayes.com/experiments/cube-3d/	CSS3 demo allows you to use the mouse to tilt and rotate the 3D cube. Includes a tutorial.
http://www.effectgames.com/demos/canvascycle/	Animated waterfall provides a nice demo of using color in HTML5 canvas.
http://macek.github.com/google_pacman/	The Google PAC-MAN® game (a Google Doodle) built in HTML5.
http://www.benjoffe.com/code/games/torus/	A 3D game similar to Tetris® built with JavaScript and canvas.
http://code.almeros.com/code-examples/water-effect-canvas/	Uses canvas and JavaScript to create a water rippling effect. Hover the cursor over the canvas to see the effect. The site includes a tutorial.
http://jqueryui.com/demos/	Numerous jQuery demos, including animations, transitions, color, interactions and more.

Fig. 1.7 | HTML5, CSS3, JavaScript, canvas and jQuery demos.
(Part 3 of 4.)



URL	Description
http://lab.smashup.it/flip/	Demonstrates a flip box using jQuery.
http://tutorialzine.com/2010/09/html5-canvas-slideshow-jquery/	Slideshow built with HTML5 canvas and jQuery (includes a tutorial).
http://css-tricks.com/examples/Circulate/	Learn how to create an animated circulation effect using jQuery.
http://demo.tutorialzine.com/2010/02/photo-shoot-css-jquery/demo.html	Uses jQuery and CSS to create a photoshoot effect, allowing you to focus on an area of the page and snap a picture (includes a tutorial).

Fig. 1.7 | HTML5, CSS3, JavaScript, canvas and jQuery demos.
(Part 4 of 4.)



1.5 Evolution of the Internet and World Wide Web

- ▶ The Internet—a global network of computers—was made possible by the *convergence of computing and communications technologies*.
- ▶ In the late 1960s, ARPA (the Advanced Research Projects Agency) rolled out blueprints for networking the main computer systems of about a dozen ARPA–funded universities and research institutions.
- ▶ They were to be connected with communications lines operating at a then–stunning 56 Kbps (i.e., 56,000 bits per second)—this at a time when most people (of the few who could) were connecting over telephone lines to computers at a rate of 110 bits per second.



1.5 Evolution of the Internet and World Wide Web (cont.)

- ▶ A **bit** (short for “binary digit”) is the smallest data item in a computer; it can assume the value 0 or 1.
- ▶ ARPA proceeded to implement the **ARPANET**, which eventually evolved into today’s **Internet**.
- ▶ Rather than enabling researchers to share each other’s computers, it rapidly became clear that communicating quickly and easily via electronic mail was the key early benefit of the ARPANET.
- ▶ This is true even today on the Internet, which facilitates communications of all kinds among the world’s Internet users.



1.5 Evolution of the Internet and World Wide Web (cont.)

Packet Switching

- ▶ One of the primary goals for ARPANET was to allow *multiple* users to send and receive information simultaneously over the *same* communications paths (e.g., phone lines).
- ▶ The network operated with a technique called **packet switching**, in which digital data was sent in small bundles called **packets**.
- ▶ The packets contained *address*, *error-control* and *sequencing* information.
- ▶ The address information allowed packets to be *routed* to their destinations.



1.5 Evolution of the Internet and World Wide Web (cont.)

- ▶ The sequencing information helped in reassembling the packets—which, because of complex routing mechanisms, could actually arrive out of order—into their original order for presentation to the recipient.
- ▶ Packets from different senders were intermixed on the same lines to efficiently use the available bandwidth.
- ▶ The network was designed to operate without centralized control.
- ▶ If a portion of the network failed, the remaining working portions would still route packets from senders to receivers over alternative paths for reliability.



1.5 Evolution of the Internet and World Wide Web (cont.)

TCP/IP

- ▶ The protocol (i.e., set of rules) for communicating over the ARPANET became known as **TCP**—the **Transmission Control Protocol**.
- ▶ TCP ensured that messages were properly routed from sender to receiver and that they arrived intact.
- ▶ As the Internet evolved, organizations worldwide were implementing their own networks for both intraorganization (i.e., within the organization) and interorganization (i.e., between organizations) communications.
- ▶ One challenge was to get these different networks to communicate.



1.5 Evolution of the Internet and World Wide Web (cont.)

- ▶ ARPA accomplished this with the development of IP—the **Internet Protocol**, truly creating a **network of networks**, the current architecture of the Internet.
- ▶ The combined set of protocols is now commonly called **TCP/IP**.
- ▶ Each computer on the Internet has a unique **IP address**.
- ▶ The current IP standard, Internet Protocol version 4 (IPv4), has been in use since 1984 and will soon run out of possible addresses.
- ▶ **IPv6** is just starting to be deployed. It features enhanced security and a new addressing scheme, hugely expanding the number of IP addresses available so that we will not run out of IP addresses in the foreseeable future.



1.5 Evolution of the Internet and World Wide Web (cont.)

Explosive Growth

- ▶ Initially, Internet use was limited to universities and research institutions; then the military began using it intensively.
- ▶ Eventually, the government decided to allow access to the Internet for commercial purposes.
- ▶ **Bandwidth** (i.e., the information-carrying capacity) on the Internet's is increasing rapidly as costs dramatically decline.



1.5 Evolution of the Internet and World Wide Web (cont.)

World Wide Web, HTML, HTTP

- ▶ The **World Wide Web** allows computer users to execute web-based applications and to locate and view multimedia-based documents on almost any subject over the Internet.
- ▶ In 1989, **Tim Berners-Lee** of CERN (the European Organization for Nuclear Research) began to develop a technology for sharing information via hyperlinked text documents.
- ▶ Berners-Lee called his invention the **HyperText Markup Language (HTML)**.
- ▶ He also wrote communication protocols to form the backbone of his new information system, which he called the World Wide Web.
- ▶ In particular, he wrote the **Hypertext Transfer Protocol (HTTP)**—a communications protocol used to send information over the web.
- ▶ The **URL (Uniform Resource Locator)** specifies the address (i.e., location) of the web page displayed in the browser window.
- ▶ Each web page on the Internet is associated with a unique URL.
- ▶ URLs usually begin with `http://`.



1.5 Evolution of the Internet and World Wide Web (cont.)

HTTPS

- ▶ URLs of websites that handle private information, such as credit card numbers, often begin with `https://`, the abbreviation for **Hypertext Transfer Protocol Secure (HTTPS)**.
- ▶ HTTPS is the standard for transferring encrypted data on the web.
- ▶ It combines HTTP with the Secure Sockets Layer (SSL) and the more recent Transport Layer Security (TLS) cryptographic schemes for securing communications and identification information over the web.



1.5 Evolution of the Internet and World Wide Web (cont.)

Mosaic, Netscape, Emergence of Web 2.0

- ▶ Web use exploded with the availability in 1993 of the Mosaic browser, which featured a user-friendly graphical interface.
- ▶ Marc Andreessen, whose team at the National Center for Supercomputing Applications (NCSA) developed Mosaic, went on to found Netscape, the company that many people credit with igniting the explosive Internet economy of the late 1990s.
- ▶ But the “dot com” economic bust brought hard times in the early 2000s.
- ▶ The resurgence that began in 2004 or so has been named **Web 2.0**.

1.6 Web Basics

- ▶ In its simplest form, a *web page* is nothing more than an HTML (HyperText Markup Language) document (with the extension `.html` or `.htm`) that describes to a web browser the document's content and structure.

Hyperlinks

- ▶ HTML documents normally contain **hyperlinks**, which, when clicked, load a specified web document.
- ▶ Both images and text may be hyperlinked.
- ▶ When the user clicks a hyperlink, a **web server** locates the requested web page and sends it to the user's web browser.
- ▶ Similarly, the user can type the *address of a web page* into the browser's *address field* and press *Enter* to view the specified page.

1.6 Web Basics (cont.)

- ▶ Hyperlinks can reference other web pages, e-mail addresses, files and more.
- ▶ If a hyperlink's URL is in the form `mailto:emailAddress`, clicking the link loads your default e-mail program and opens a **message window** addressed to the specified e-mail address.
- ▶ If a hyperlink references a file that the browser is incapable of displaying, the browser prepares to **download** the file, and generally prompts the user for information about how the file should be stored.

1.6 Web Basics (cont.)

URIs and URLs

- ▶ *URIs (Uniform Resource Identifiers)* identify resources on the Internet.
- ▶ URIs that start with `http://` are called *URLs (Uniform Resource Locators)*.

Parts of a URL

- ▶ A URL contains information that directs a browser to the resource that the user wishes to access.
- ▶ Web servers make such resources available to web clients.
- ▶ Popular web servers include Apache's HTTP Server and Microsoft's Internet Information Services (IIS).



1.6 Web Basics (cont.)

- ▶ Let's examine the components of the URL
 - `http://www.deitel.com/books/downloads.html`
- ▶ The text `http://` indicates that the HyperText Transfer Protocol (HTTP) should be used to obtain the resource.
- ▶ Next in the URL is the server's fully qualified **hostname** (for example, `www.deitel.com`)—the name of the web-server computer on which the resource resides.
- ▶ This computer is referred to as the **host**, because it houses and maintains resources.
- ▶ The hostname `www.deitel.com` is translated into an **IP (Internet Protocol) address**—a numerical value that uniquely identifies the server on the Internet.
- ▶ An Internet **Domain Name System (DNS)** server maintains a database of hostnames and their corresponding IP addresses and performs the translations automatically.

1.6 Web Basics (cont.)

- ▶ The remainder of the URL (/books/downloads.html) specifies the resource's location (/books) and name (downloads.html) on the web server.
- ▶ The location could represent an actual directory on the web server's file system. For *security* reasons, however, the location is typically a *virtual directory*.
- ▶ The web server translates the virtual directory into a real location on the server, thus hiding the resource's true location.

Making a Request and Receiving a Response

- ▶ Figure 1.8 shows a web browser sending a request to a web server.

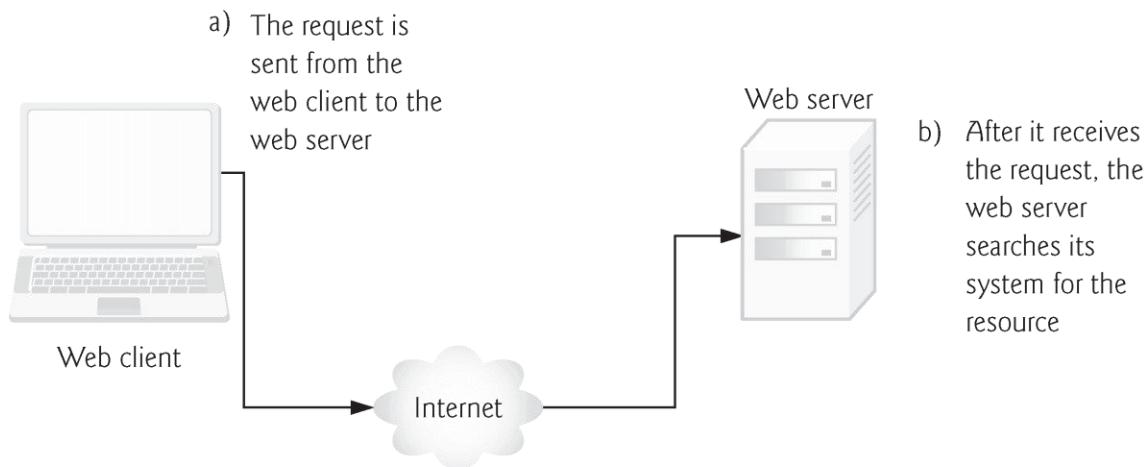


Fig. 1.8 | Client requesting a resource from a web server.

1.6 Web Basics (cont.)

- ▶ In Fig. 1.8, the web browser sends an HTTP request to the server. The request (in its simplest form) is
 - GET /books/downloads.html HTTP/1.1
- ▶ The word **GET** is an **HTTP method** indicating that the client wishes to obtain a resource from the server.
- ▶ The remainder of the request provides the path name of the resource (e.g., an HTML5 document) and the protocol's name and version number (HTTP/1.1).
- ▶ The client's request also contains some required and optional headers.
- ▶ Figure 1.9 shows the web server responding to a request.

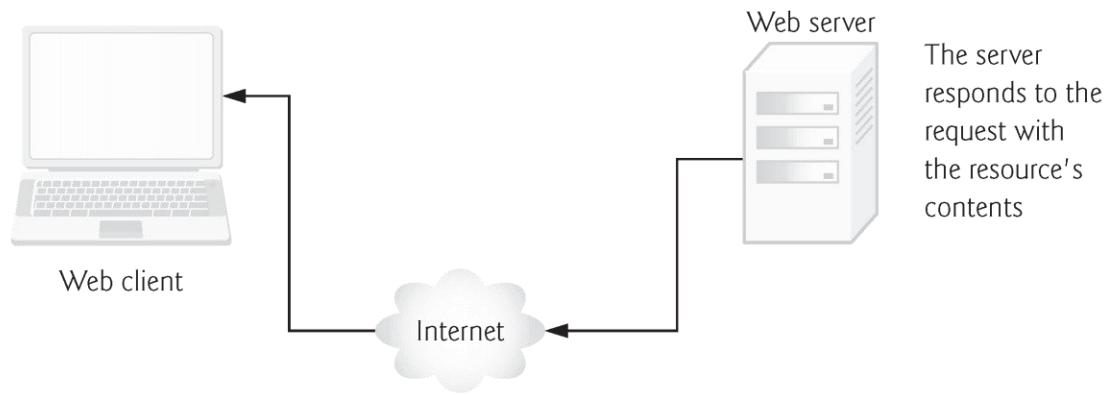


Fig. 1.9 | Client receiving a response from the web server.



1.6 Web Basics (cont.)

- ▶ The server first sends a line of text that indicates the HTTP version, followed by a numeric code and a phrase describing the status of the transaction. For example,
 - HTTP/1.1 200 OK
- ▶ indicates success, whereas
 - HTTP/1.1 404 Not found
- ▶ informs the client that the web server could not locate the requested resource.

HTTP Headers

- ▶ Next, the server sends one or more **HTTP headers**, which provide additional information about the data that will be sent.
- ▶ In this case, the server is sending an HTML5 text document, so one HTTP header for this example would read:
 - Content-type: text/html

1.6 Web Basics (cont.)

- ▶ The information provided in this header specifies the **Multipurpose Internet Mail Extensions (MIME) type** of the content that the server is transmitting to the browser.
- ▶ The MIME standard specifies data formats, which programs can use to interpret data correctly.
- ▶ For example, the MIME type `text/plain` indicates that the sent information is text that can be displayed directly.
- ▶ Similarly, the MIME type `image/jpeg` indicates that the content is a JPEG image.
- ▶ When the browser receives this MIME type, it attempts to display the image.

1.6 Web Basics (cont.)

- ▶ The header or set of headers is followed by a blank line, which indicates to the client browser that the server is finished sending HTTP headers.
- ▶ Finally, the server sends the contents of the requested document (downloads.html).
- ▶ The client-side browser then renders (or displays) the document, which may involve additional HTTP requests to obtain associated CSS and images.

1.6 Web Basics (cont.)

HTTP get and post Requests

- ▶ The two most common **HTTP request types** (also known as **request methods**) are get and post.
- ▶ A get request typically gets (or retrieves) information from a server, such as an HTML document, an image or search results based on a user-submitted search term.
- ▶ A post request typically posts (or sends) data to a server.
- ▶ Common uses of post requests are to send form data or documents to a server.
- ▶ An HTTP request often posts data to a **server-side form handler** that processes the data.
- ▶ For example, when a user performs a search or participates in a web-based survey, the web server receives the information specified in the HTML form as part of the request. Get requests and post requests can both be used to send data to a web server, but each request type sends the information differently.



1.6 Web Basics (cont.)

- ▶ A get request appends data to the URL, e.g., `www.google.com/search?q=deitel`.
- ▶ In this case `search` is the name of Google's server-side form handler, `q` is the name of a variable in Google's search form and `deitel` is the search term.
- ▶ The `?` in the preceding URL separates the **query string** from the rest of the URL in a request.
- ▶ A *name/value* pair is passed to the server with the *name* and the *value* separated by an equals sign (`=`).
- ▶ If more than one *name/value* pair is submitted, each pair is separated by an ampersand (`&`).
- ▶ The server uses data passed in a query string to retrieve an appropriate resource from the server.
- ▶ The server then sends a response to the client. A get request may be initiated by submitting an HTML form whose `method` attribute is set to "get", or by typing the URL (possibly containing a query string) directly into the browser's address bar.

1.6 Web Basics (cont.)

- ▶ A post request sends form data as part of the HTTP message, not as part of the URL.
- ▶ A get request typically limits the query string (i.e., everything to the right of the ?) to a specific number of characters, so it's often necessary to send large amounts of information using the post method.
- ▶ The post method is also sometimes preferred because it hides the submitted data from the user by embedding it in an HTTP message.
- ▶ If a form submits several hidden input values along with user-submitted data, the post method might generate a URL like www.searchengine.com/search.
- ▶ The form data still reaches the server and is processed in a similar fashion to a get request, but the user does not see the exact information sent.



Software Engineering Observation 1.1

The data sent in a post request is not part of the URL, and the user can't see the data by default. However, tools are available that expose this data, so you should not assume that the data is secure just because a post request is used.

1.6 Web Basics (cont.)

Client-Side Caching

- ▶ Browsers often **cache** (save on disk) recently viewed web pages for quick reloading.
- ▶ If there are no changes between the version stored in the cache and the current version on the web, this speeds up your browsing experience.
- ▶ An HTTP response can indicate the length of time for which the content remains “fresh.”
- ▶ If this amount of time has not been reached, the browser can avoid another request to the server. If not, the browser loads the document from the cache.
- ▶ Similarly, there’s also the “not modified” HTTP response, indicating that the file content has not changed since it was last requested (which is information that’s send in the request).
- ▶ Browsers typically do not cache the server’s response to a post request, because the next post might not return the same result.

1.7 Multitier Application Architecture

- ▶ Web-based applications are often **multitier applications** (sometimes referred to as *n-tier applications*) that divide functionality into separate **tiers** (i.e., logical groupings of functionality).
- ▶ Figure 1.10 presents the basic structure of a **three-tier web-based application**.

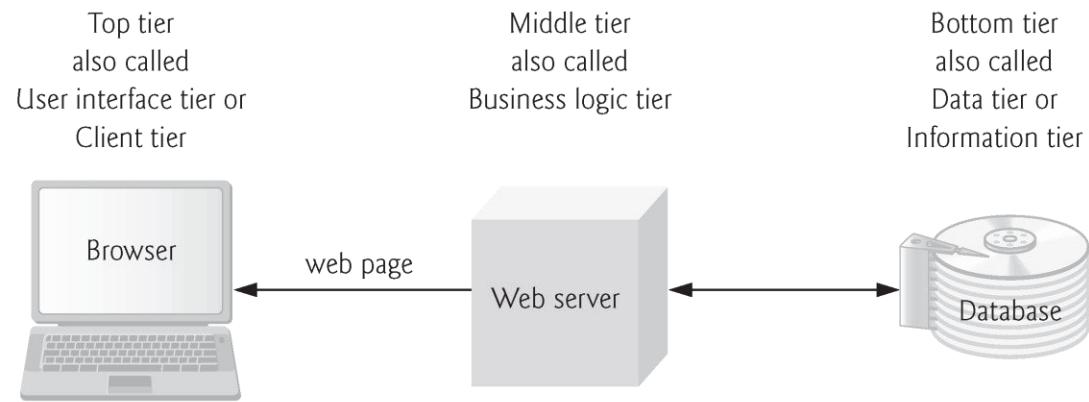


Fig. 1.10 | Three-tier architecture.



1.7 Multitier Application Architecture (cont.)

- ▶ The **bottom tier** (also called the data tier or the information tier) maintains the application's data.
- ▶ This tier typically stores data in a relational database management system (RDBMS).
- ▶ The **middle tier** implements business logic, controller logic and presentation logic to control interactions between the application's clients and its data.
- ▶ The middle tier acts as an intermediary between data in the information tier and the application's clients.
- ▶ The middle-tier **controller logic** processes client requests (such as requests to view a product catalog) and retrieves data from the database.
- ▶ The middle-tier **presentation logic** then processes data from the information tier and presents the content to the client.
- ▶ Web applications typically present data to clients as HTML documents.



1.7 Multitier Application Architecture (cont.)

- ▶ **Business logic** in the middle tier enforces **business rules** and ensures that data is reliable before the application updates a database or presents data to users.
- ▶ Business rules dictate how clients access data and how applications process data.
- ▶ The **top tier**, or client tier, is the application's user interface, which gathers input and displays output.
- ▶ Users interact directly with the application through the user interface, which is typically a web browser or a mobile device.
- ▶ In response to user actions (e.g., clicking a hyperlink), the client tier interacts with the middle tier to make requests and to retrieve data from the information tier.
- ▶ The client tier then displays the data retrieved for the user.



1.8 Client-Side Scripting versus Server-Side Scripting

- ▶ Client-side scripting with JavaScript can be used to validate user input, to interact with the browser, to enhance web pages, and to add client/server communication between a browser and a web server.
- ▶ Client-side scripting does have limitations, such as browser dependency; the browser or **scripting host** must support the scripting language and capabilities.
- ▶ Scripts are restricted from arbitrarily accessing the local hardware and file system for security reasons.
- ▶ Another issue is that client-side scripts can be viewed by the client by using the browser's source-viewing capability.
- ▶ Sensitive information, such as passwords or other personally identifiable data, should not be on the client.
- ▶ All client-side data validation should be mirrored on the server. Also, placing certain operations in JavaScript on the client can open web applications to security issues.



1.8 Client-Side Scripting versus Server-Side Scripting (cont.)

- ▶ Programmers have more flexibility with **server-side scripts**, which often generate custom responses for clients.
- ▶ For example, a client might connect to an airline's web server and request a list of flights from Boston to San Francisco between April 19 and May 5. The server queries the database, dynamically generates an HTML document containing the flight list and sends the document to the client. This technology allows clients to obtain the most current flight information from the database by connecting to an airline's web server.
- ▶ Server-side scripting languages have a wider range of programmatic capabilities than their client-side equivalents.
- ▶ Server-side scripts also have access to server-side software that extends server functionality—Microsoft web servers use **ISAPI (Internet Server Application Program Interface) extensions** and Apache HTTP Servers use **modules**.

1.9 World Wide Web Consortium (W3C)

- ▶ In October 1994, Tim Berners-Lee founded an organization—the [World Wide Web Consortium \(W3C\)](#)—devoted to developing nonproprietary, interoperable technologies for the World Wide Web.
- ▶ One of the W3C's primary goals is to make the web universally *accessible*—regardless of disability, language or culture.
- ▶ The W3C is also a standards organization.
- ▶ Web technologies standardized by the W3C are called [Recommendations](#).
- ▶ Current and forthcoming W3C Recommendations include the HyperText Markup Language 5 (HTML5), Cascading Style Sheets 3 (CSS3) and the Extensible Markup Language (XML).

1.10 Web 2.0: Going Social

- ▶ In 2003 there was a noticeable shift in how people and businesses were using the web and developing web-based applications.
- ▶ The term **Web 2.0** was coined by **Dale Dougherty** of **O'Reilly Media** in 2003 to describe this trend.
 - T. O'Reilly, "What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software." September 2005
<http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html?page=1>.
- ▶ Generally, Web 2.0 companies use the web as a platform to create collaborative, community-based sites (e.g., social networking sites, blogs, wikis).

1.10 Web 2.0: Going Social (cont.)

Web 1.0 versus Web 2.0

- ▶ **Web 1.0** (the state of the web through the 1990s and early 2000s) was focused on a relatively small number of companies and advertisers producing content for users to access (some people called it the “brochure web”).
- ▶ Web 2.0 *involves* the users—not only do they often create content, but they help organize it, share it, remix it, critique it, update it, etc.
- ▶ One way to look at Web 1.0 is as a *lecture*, a small number of professors informing a large audience of students. In comparison, Web 2.0 is a *conversation*, with everyone having the opportunity to speak and share views.



1.10 Web 2.0: Going Social (cont.)

Architecture of Participation

- ▶ Web 2.0 embraces an **architecture of participation**—a design that encourages user interaction and community contributions.
- ▶ The architecture of participation has influenced software development as well.
- ▶ Opensource software is available for anyone to use and modify with few or no restrictions (we'll say more about open source in Section 1.12).
- ▶ Using **collective intelligence**—the concept that a large diverse group of people will create smart ideas—communities collaborate to develop software that many people believe is better and more robust than proprietary software.
- ▶ Rich Internet Applications (RIAs) are being developed using technologies (such as Ajax) that have the look and feel of desktop software, enhancing a user's overall experience.

1.10 Web 2.0: Going Social (cont.)

Search Engines and Social Media

- ▶ The way we find the information on these sites is also changing—people are **tagging** (i.e., labeling) web content by subject or keyword in a way that helps anyone locate information more effectively.
 - *Semantic Web*
- ▶ In the future, computers will learn to understand the meaning of the data on the web—the beginnings of the **Semantic Web** are already appearing.

1.10 Web 2.0: Going Social (cont.)

Google

- ▶ In 1996, Stanford computer science Ph.D. candidates Larry Page and Sergey Brin began collaborating on a new search engine.
- ▶ In 1997, they chose the name Google—a play on the mathematical term *googol*, a quantity represented by the number “one” followed by 100 “zeros” (or 10^{100})—a staggeringly large number.
- ▶ Google’s ability to return extremely accurate search results quickly helped it become the most widely used search engine and one of the most popular websites in the world.

1.10 Web 2.0: Going Social (cont.)

Web Services and Mashups

- ▶ We include in this book a substantial treatment of web services and introduce the applications-development methodology of *mashups*, in which you can rapidly develop powerful and intriguing applications by combining (often free) complementary web services and other forms of information feeds (Fig. 1.11).

Web services source	How it's used
Google Maps	Mapping services
Facebook	Social networking
Foursquare	Mobile check-in
LinkedIn	Social networking for business
YouTube	Video search
Twitter	Microblogging
Groupon	Social commerce
Netflix	Movie rentals
eBay	Internet auctions
Wikipedia	Collaborative encyclopedia
PayPal	Payments
Last.fm	Internet radio

Fig. 1.11 | Some popular web services that you can use to build web applications (www.programmableweb.com/apis/directory/1?sort=mashups). (Part I of 2.)

Web services source	How it's used
Amazon eCommerce	Shopping for books and more
Salesforce.com	Customer Relationship Management (CRM)
Skype	Internet telephony
Microsoft Bing	Search
Flickr	Photo sharing
Zillow	Real estate pricing
Yahoo Search	Search
WeatherBug	Weather

Fig. 1.11 | Some popular web services that you can use to build web applications (www.programmableweb.com/apis/directory/1?sort=mashups). (Part 2 of 2.)

1.10 Web 2.0: Going Social (cont.)

- ▶ Web services, inexpensive computers, abundant high-speed Internet access, open source software and many other elements have inspired new, exciting, *lightweight business models* that people can launch with only a small investment.
- ▶ Some types of websites with rich and robust functionality that might have required hundreds of thousands or even millions of dollars to build in the 1990s can now be built for nominal sums.

1.10 Web 2.0: Going Social (cont.)

Ajax

- ▶ **Ajax** is one of the premier Web 2.0 software technologies (Fig. 1.12).
- ▶ Ajax helps Internet-based applications perform like desktop applications—a difficult task, given that such applications suffer transmission delays as data is shuttled back and forth between your computer and servers on the Internet.

1.10 Web 2.0: Going Social (cont.)

Social Applications

- ▶ Figure 1.13 discusses a few of the social applications that are making an impact.



Company	Description
Facebook	<p>Facebook was launched in 2004 and is already worth an estimated \$100 billion. By January 2011, Facebook was the most active site on the Internet with more than 750 million users who were spending 700 billion minutes on Facebook per month (www.facebook.com/press/info.php?statistics). At its current growth rate (about 5% per month), Facebook will reach one billion users in 2012, out of two billion Internet users! The activity on the site makes it extremely attractive for application developers. Each day, over 20 million applications are installed by Facebook users (www.facebook.com/press/info.php?statistics).</p>
Twitter	<p>Twitter (founded in 2006) has revolutionized <i>microblogging</i>. Users post tweets—messages up to 140 characters long. Approximately 140 million tweets are posted per day. You can follow the tweets of friends, celebrities, businesses, government representatives (including Barack Obama, who has 10 million followers), and so on, or you can follow tweets by subject to track news, trends and more. At the time of this writing, Lady Gaga had the most followers (over 13 million). Twitter has become the point of origin for many breaking news stories worldwide.</p>

Fig. 1.13 | Social applications. (Part 1 of 4.)

Company	Description
Groupon	<p>Groupon, a <i>social commerce</i> site, was launched in 2008. By August 2011 the company was valued as high as \$25 billion, making it the fastest growing company ever! Groupon offers daily deals in each market for restaurants, retailers, services, attractions and more. Deals are activated only after a minimum number of people sign up to buy the product or service. If you sign up for a deal and it has yet to meet the minimum, you might be inclined to tell others about the deal via e-mail, Facebook, Twitter, etc. One of the most successful national Groupon deals to date was a certificate for \$50 worth of merchandise from a major retailer for \$25. More than 620,000 vouchers were sold in one day (www.huffingtonpost.com/2011/06/30/the-most-successful-group_n_887711.html)!</p>

Fig. 1.13 | Social applications. (Part 2 of 4.)



Company	Description
Foursquare	<p>Foursquare, launched in 2009, is a mobile <i>check-in</i> application that allows you to notify your friends of your whereabouts. You can download the app to your smartphone and link it to your Facebook and Twitter accounts so your friends can follow you from multiple platforms. If you do not have a smartphone, you can check in by text message. Foursquare uses GPS to determine your location. Businesses use Foursquare to send offers to users in the area. Launched in March 2009, Foursquare already has over 10 million users worldwide (foursquare.com/about).</p>
Skype	<p>Skype (founded in 2003) allows you to make mostly free voice and video calls over the Internet using a technology called <i>VoIP</i> (<i>Voice over IP</i>; IP stands for “Internet Protocol”). The company was recently sold to Microsoft for \$8.5 billion.</p>

Fig. 1.13 | Social applications. (Part 3 of 4.)

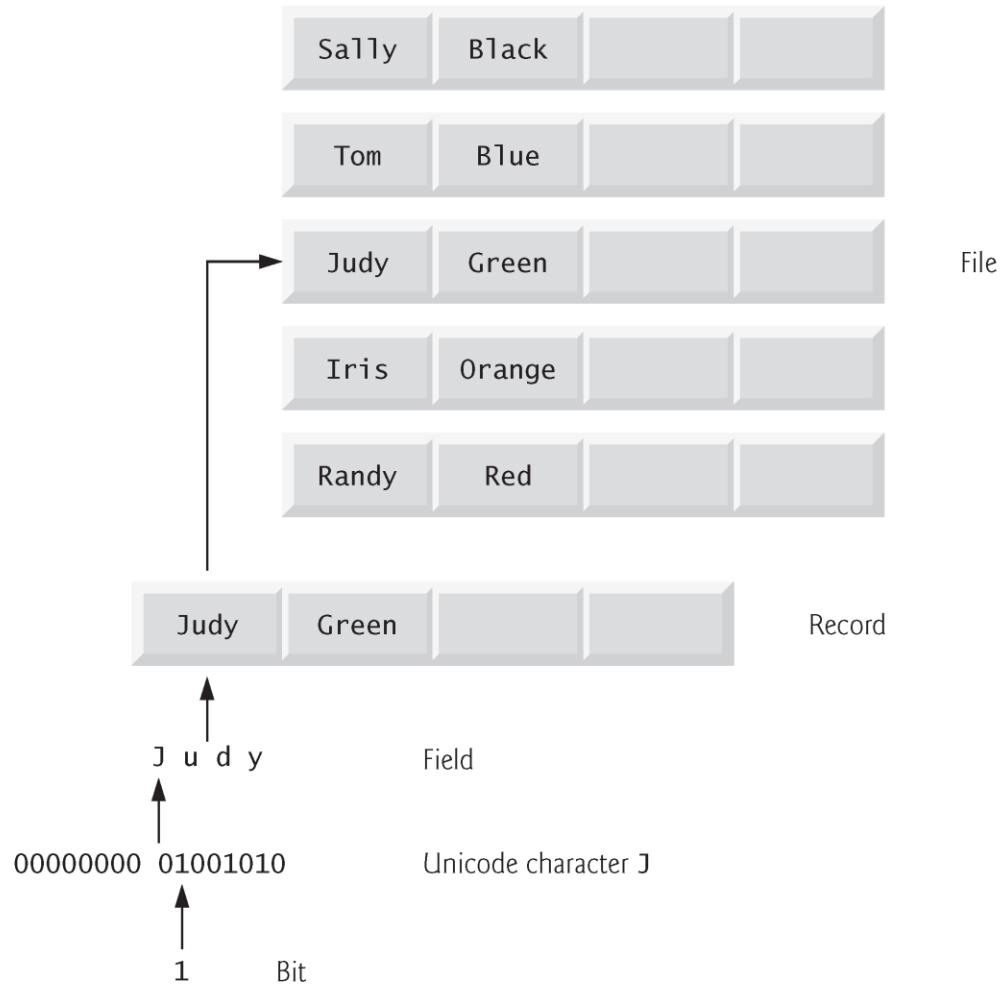


Company	Description
YouTube	YouTube is a video-sharing site that was founded in 2005. Within one year, the company was purchased by Google for \$1.65 billion. YouTube now accounts for 8.2% of all Internet traffic (www.engadget.com/2011/05/17/study-finds-netflix-is-the-largest-source-of-internet-traffic-in/). Within one week of the release of Apple's iPhone 3GS—the first iPhone model to offer video—mobile uploads to YouTube grew 400% (www.hypebot.com/hypebot/2009/06/youtube-reports-1700-jump-in-mobile-video.html).

Fig. 1.13 | Social applications. (Part 4 of 4.)

1.11 Data Hierarchy

- ▶ Data items processed by computers form a **data hierarchy** that becomes larger and more complex in structure as we progress from bits to characters to fields, and so on.
- ▶ Figure 1.14 illustrates a portion of the data hierarchy.
- ▶ Figure 1.15 summarizes the data hierarchy's levels.





Level	Description
Bits	<p>The smallest data item in a computer can assume the value 0 or the value 1. Such a data item is called a bit (short for “binary digit”—a digit that can assume one of two values). It’s remarkable that the impressive functions performed by computers involve only the simplest manipulations of 0s and 1s—<i>examining a bit’s value, setting a bit’s value and reversing a bit’s value</i> (from 1 to 0 or from 0 to 1).</p>
Characters	<p>It’s tedious for people to work with data in the low-level form of bits. Instead, they prefer to work with <i>decimal digits</i> (0–9), <i>letters</i> (A–Z and a–z), and <i>special symbols</i> (e.g., \$, @, %, &, *, (,), –, +, ", :, ? and /). Digits, letters and special symbols are known as characters. The computer’s character set is the set of all the characters used to write programs and represent data items. Computers process only 1s and 0s, so a computer’s character set represents every character as a pattern of 1s and 0s. Java uses Unicode® characters that are composed of two bytes, each composed of eight bits. Unicode contains characters for many of the world’s languages. See Appendix F for more information on Unicode. See Appendix D for more information on the ASCII (American Standard Code for Information Interchange) character set—the popular subset of Unicode that represents uppercase and lowercase letters, digits and some common special characters.</p>

Fig. 1.15 | Levels of the data hierarchy. (Part 1 of 4.)

Level	Description
Fields	Just as characters are composed of bits, fields are composed of characters or bytes. A field is a group of characters or bytes that conveys meaning. For example, a field consisting of uppercase and lowercase letters could be used to represent a person's name, and a field consisting of decimal digits could represent a person's age.

Fig. 1.15 | Levels of the data hierarchy. (Part 2 of 4.)



Level	Description
Records	<p>Several related fields can be used to compose a record (implemented as a <code>class</code> in Java). In a payroll system, for example, the record for an employee might consist of the following fields (possible types for these fields are shown in parentheses):</p> <ul style="list-style-type: none">• Employee identification number (a whole number)• Name (a string of characters)• Address (a string of characters)• Hourly pay rate (a number with a decimal point)• Year-to-date earnings (a number with a decimal point)• Amount of taxes withheld (a number with a decimal point) <p>Thus, a record is a group of related fields. In the preceding example, all the fields belong to the same employee. A company might have many employees and a payroll record for each one.</p>

Fig. 1.15 | Levels of the data hierarchy. (Part 3 of 4.)



Level	Description
Files	<p>A file is a group of related records. [Note: More generally, a file contains arbitrary data in arbitrary formats. In some operating systems, a file is viewed simply as a <i>sequence of bytes</i>—any organization of the bytes in a file, such as organizing the data into records, is a view created by the application programmer.] It's not unusual for an organization to have many files, some containing billions, or even trillions, of characters of information.</p>
Database	<p>A database is an electronic collection of data that's organized for easy access and manipulation. There are various database models. In this book, we introduce relational databases in which data is stored in simple <i>tables</i>. A table includes <i>records</i> and <i>fields</i>. For example, a table of students might include first name, last name, major, year, student ID number and grade point average. The data for each student is a record, and the individual pieces of information in each record are the fields. You can search, sort and manipulate the data based on its relationship to multiple tables or databases. For example, a university might use data from the student database in combination with databases of courses, on-campus housing, meal plans, etc. We discuss databases in Chapter 18 and use them in the server-side programming chapters.</p>

Fig. 1.15 | Levels of the data hierarchy. (Part 4 of 4.)

1.12 Operating Systems

- ▶ **Operating systems** are software systems that make using computers more convenient for users, application developers and system administrators.
- ▶ Operating systems provide services that allow each application to execute safely, efficiently and *concurrently* (i.e., in parallel) with other applications.
- ▶ The software that contains the core components of the operating system is called the **kernel**.
- ▶ Popular desktop operating systems include Linux, Windows 7 and Mac OS X.
- ▶ Popular mobile operating systems used in smartphones and tablets include Google's Android, Apple's iOS (for iPhone, iPad and iPod Touch devices), BlackBerry OS and Windows Phone 7.



1.12.1 Desktop and Notebook Operating Systems

- ▶ In this section we discuss two of the popular desktop operating systems—the proprietary Windows operating system and the open source Linux operating system.

Windows—A Proprietary Operating System

- ▶ In the mid-1980s, Microsoft developed the [Windows operating system](#), consisting of a graphical user interface built on top of DOS—an enormously popular personal-computer operating system of the time that users interacted with by *typing* commands.
- ▶ Windows borrowed from many concepts (such as icons, menus and windows) developed by Xerox PARC and popularized by early Apple Macintosh operating systems.
- ▶ Windows is a *proprietary* operating system—it's controlled by Microsoft exclusively.



1.12.1 Desktop and Notebook Operating Systems (cont.)

Linux—An Open-Source Operating System

- ▶ The Linux operating system is perhaps the greatest success of the *open-source* movement.
- ▶ **Open-source software** departs from the *proprietary* software development style that dominated software's early years.
- ▶ With open-source development, individuals and companies *contribute* their efforts in developing, maintaining and evolving software in exchange for the right to use that software for their own purposes, typically at no charge.
- ▶ Rapid improvements to computing and communications, decreasing costs and open-source software have made it much easier and more economical to create a software-based business now than just a decade ago.
- ▶ A great example is Facebook, which was launched from a college dorm room and built with open-source software.
 - developers.facebook.comopensource/.



1.12.1 Desktop and Notebook Operating Systems (cont.)

- ▶ The [Linux](#) kernel is the core of the most popular open-source, freely distributed, full-featured operating system.
- ▶ It's developed by a loosely organized team of volunteers and is popular in servers, personal computers and embedded systems.
- ▶ Unlike that of proprietary operating systems like Microsoft's Windows and Apple's Mac OS X, Linux source code (the program code) is available to the public for examination and modification and is free to download and install.
- ▶ Linux has become extremely popular on servers and in embedded systems, such as Google's Android-based smartphones.

1.12.2 Mobile Operating Systems

- ▶ Two of the most popular mobile operating systems are Apple's iOS and Google's Android.
- ▶ Apple's proprietary iPhone operating system, **iOS**, is derived from Apple's Mac OS X and is used in the iPhone, iPad and iPod Touch devices.
- ▶ **Android**—the fastest growing mobile and smartphone operating system—is based on the Linux kernel and Java.
- ▶ One benefit of developing Android apps is the openness of the platform. The operating system is open source and free.

1.13 Types of Programming Languages

- ▶ Programmers write instructions in various programming languages, some directly understandable by computers and others requiring intermediate *translation* steps.
- ▶ Any computer can directly understand only its own **machine language**, defined by its hardware design.
- ▶ Machine languages generally consist of numbers (ultimately reduced to 1s and 0s). Such languages are cumbersome for humans.
- ▶ Programming in machine language—the numbers that computers could directly understand—was simply too slow and tedious for most programmers.
- ▶ Instead, they began using Englishlike abbreviations to represent elementary operations.
- ▶ These abbreviations formed the basis of **assembly languages**.
- ▶ *Translator programs* called **assemblers** were developed to convert assembly-language programs to machine language.



1.13 Types of Programming Languages (cont.)

- ▶ Although assembly-language code is clearer to humans, it's incomprehensible to computers until translated to machine language.
- ▶ To speed the programming process even further, **high-level languages** were developed in which single statements could be written to accomplish substantial tasks.
- ▶ High-level languages allow you to write instructions that look almost like everyday English and contain commonly used mathematical expressions.
- ▶ Translator programs called **compilers** convert high-level language programs into machine language.
- ▶ **Interpreter** programs were developed to execute high-level language programs directly, although more slowly than compiled programs.
- ▶ Figure 1.16 introduces a number of popular programming languages.
- ▶



Performance Tip 1.1

Interpreters have an advantage over compilers in Internet scripting. An interpreted program can begin executing as soon as it's downloaded to the client's machine, without needing to be compiled before it can execute. On the downside, interpreted scripts generally run slower than compiled code.



Programming language	Description
C	C was implemented in 1972 by Dennis Ritchie at Bell Laboratories. It initially became widely known as the UNIX operating system's development language. Today, most of the code for general-purpose operating systems is written in C or C++.
C++	C++, an extension of C, was developed by Bjarne Stroustrup in the early 1980s at Bell Laboratories. C++ provides a number of features that "spruce up" the C language, but more important, it provides capabilities for object-oriented programming.
Objective-C	Objective-C is an object-oriented language based on C. It was developed in the early 1980s and later acquired by NeXT, which in turn was acquired by Apple. It has become the key programming language for the Mac OS X operating system and all iOS-based devices (such as iPods, iPhones and iPads).
Visual Basic	Microsoft's Visual Basic language (based on the Basic language developed at Dartmouth College in the 1960s) was introduced in the early 1990s to simplify Microsoft Windows applications development. Its latest versions support object-oriented programming.

Fig. 1.16 | Popular programming languages. (Part 1 of 5.)



Programming language	Description
Visual C#	Microsoft's three primary object-oriented programming languages are Visual Basic, Visual C++ (based on C++) and C# (based on C++ and Java, and developed for integrating the Internet and the web into computer applications).
Java	Sun Microsystems in 1991 funded an internal corporate research project led by James Gosling, which resulted in the C++-based object-oriented programming language called Java. A key goal of Java is to enable the writing of programs that will run on a great variety of computer systems and computer-controlled devices. This is sometimes called "write once, run anywhere." Java is used to develop large-scale enterprise applications, to enhance the functionality of web servers (the computers that provide the content we see in our web browsers), to provide applications for consumer devices (smartphones, television set-top boxes and more) and for many other purposes.

Fig. 1.16 | Popular programming languages. (Part 2 of 5.)



Programming language	Description
PHP	PHP—an object-oriented, “open-source” (see Section 1.12) “scripting” language supported by a community of users and developers—is used by numerous websites including Wikipedia and Facebook. PHP is <i>platform independent</i> —implementations exist for all major UNIX, Linux, Mac and Windows operating systems. PHP also supports many databases, including MySQL. Two other popular languages similar in concept to PHP are Perl and Python. The term “LAMP” describes four key technologies for building open-source software—Linux (operating system), Apache (web server), MySQL (database) and PHP or Perl or Python (server-side scripting languages).
Python	Python, another object-oriented scripting language, was released publicly in 1991. Developed by Guido van Rossum of the National Research Institute for Mathematics and Computer Science in Amsterdam (CWI), Python draws heavily from Modula-3—a systems-programming language. Python is “extensible”—it can be extended through classes and programming interfaces.

Fig. 1.16 | Popular programming languages. (Part 3 of 5.)



Programming language	Description
JavaScript	JavaScript—developed by Brendan Eich at Netscape—is the most widely used scripting language. It's primarily used to add programmability to web pages—for example, animations and interactivity with the user. It's provided with all major web browsers.
Ruby on Rails	Ruby—created in the mid-1990s by Yukihiro Matsumoto—is an open-source, object-oriented programming language with a simple syntax that's similar to Python. Ruby on Rails combines the scripting language Ruby with the Rails web-application framework developed by 37Signals. Their book, <i>Getting Real</i> (gettingreal.37signals.com/toc.php), is a must read for web developers. Many Ruby on Rails developers have reported productivity gains over other languages when developing database-intensive web applications. Ruby on Rails was used to build Twitter's user interface.

Fig. 1.16 | Popular programming languages. (Part 4 of 5.)



Programming language	Description
Scala	Scala (www.scala-lang.org/node/273)—short for “scalable language”—was designed by Martin Odersky, a professor at École Polytechnique Fédérale de Lausanne (EPFL) in Switzerland. Released in 2003, Scala uses both the <i>object-oriented</i> and <i>functional programming</i> paradigms and is designed to integrate with Java. Programming in Scala can significantly reduce the amount of code in your applications. Twitter and Foursquare use Scala.

Fig. 1.16 | Popular programming languages. (Part 5 of 5.)

1.14 Object Technology

- ▶ *Objects*, or more precisely the *classes* objects come from, are essentially *reusable* software components.
- ▶ Almost any *noun* can be reasonably represented as a software object in terms of *attributes* (e.g., name, color and size) and *behaviors* (e.g., calculating, moving and communicating).
- ▶ Software developers are discovering that using a modular, object-oriented design and implementation approach can make software-development groups much more productive than was possible with earlier techniques—object-oriented programs are often easier to understand, correct and modify.

The Automobile as an Object

- ▶ Suppose you want to *drive a car and make it go faster by pressing its accelerator pedal*.
- ▶ Before you can drive a car, someone has to *design* it.
- ▶ A car typically begins as engineering drawings, similar to the *blueprints* that describe the design of a house.
- ▶ These drawings include the design for an accelerator pedal.

1.14 Object Technology (cont.)

- ▶ The pedal *hides* from the driver the complex mechanisms that actually make the car go faster, just as the brake pedal hides the mechanisms that slow the car, and the steering wheel *hides* the mechanisms that turn the car.
- ▶ This enables people with little or no knowledge of how engines, braking and steering mechanisms work to drive a car easily.
- ▶ Before you can drive a car, it must be *built* from the engineering drawings that describe it.
- ▶ A completed car has an *actual* accelerator pedal to make the car go faster, but even that's not enough—the car won't accelerate on its own (hopefully!), so the driver must *press* the pedal to accelerate the car.

1.14 Object Technology (cont.)

Methods and Classes

- ▶ Performing a task in a program requires a **method**.
- ▶ The method houses the program statements that actually perform its tasks.
- ▶ It hides these statements from its user, just as a car's accelerator pedal hides from the driver the mechanisms of making the car go faster.
- ▶ In object-oriented programming languages, we create a program unit called a **class** to house the set of methods that perform the class's tasks.
- ▶ For example, a class that represents a bank account might contain one method to *deposit* money to an account, another to *withdraw* money from an account and a third to *inquire* what the account's current balance is.
- ▶ A class is similar in concept to a car's engineering drawings, which house the design of an accelerator pedal, steering wheel, and so on.

1.14 Object Technology (cont.)

Instantiation

- ▶ Just as someone has to *build a car* from its engineering drawings before you can actually drive a car, you must *build an object* from a class before a program can perform the tasks that the class's methods define.
- ▶ The process of doing this is called *instantiation*. An object is then referred to as an **instance** of its class.

Reuse

- ▶ Just as a car's engineering drawings can be *reused* many times to build many cars, you can *reuse* a class many times to build many objects.
- ▶ Reuse of existing classes when building new classes and programs saves time and effort.
- ▶ Reuse also helps you build more reliable and effective systems, because existing classes and components often have gone through extensive *testing, debugging* and *performance tuning*.



Software Engineering Observation 1.2

Use a building-block approach to creating your programs. Avoid reinventing the wheel—use existing pieces wherever possible. This software reuse is a key benefit of object-oriented programming.

1.14 Object Technology (cont.)

Messages and Method Calls

- ▶ When you drive a car, pressing its gas pedal sends a *message* to the car to perform a task—that is, to go faster. Similarly, you *send messages to an object*.
- ▶ Each message is implemented as a **method call** that tells a method of the object to perform its task.
- ▶ For example, a program might call a particular bank-account object's *deposit* method to increase the account's balance.

1.14 Object Technology (cont.)

Attributes and Instance Variables

- ▶ A car, besides having capabilities to accomplish tasks, also has *attributes*, such as its color, its number of doors, the amount of gas in its tank, its current speed and its record of total miles driven (i.e., its odometer reading).
- ▶ Like its capabilities, the car's attributes are represented as part of its design in its engineering diagrams (which, for example, include an odometer and a fuel gauge).
- ▶ As you drive an actual car, these attributes are carried along with the car.
- ▶ Every car maintains its *own* attributes.
- ▶ For example, each car knows how much gas is in its own gas tank, but *not* how much is in the tanks of *other* cars.
- ▶ An object, similarly, has attributes that it carries along as it's used in a program.

1.14 Object Technology (cont.)

- ▶ These attributes are specified as part of the object's class.
- ▶ For example, a bank-account object has a *balance attribute* that represents the amount of money in the account.
- ▶ Each bank-account object knows the balance in the account it represents, but *not* the balances of the *other* accounts in the bank.
- ▶ Attributes are specified by the class's **instance variables**.

1.14 Object Technology (cont.)

Encapsulation

- ▶ Classes **encapsulate** (i.e., wrap) attributes and methods into objects—an object's attributes and methods are intimately related.
- ▶ Objects may communicate with one another, but normally they're not allowed to know how other objects are implemented—implementation details are *hidden* within the objects themselves.
- ▶ This **information hiding** is crucial to good software engineering.

1.14 Object Technology (cont.)

Inheritance

- ▶ A new class of objects can be created quickly and conveniently by **inheritance**—the new class absorbs the characteristics of an existing class, possibly customizing them and adding unique characteristics of its own.
- ▶ In our car analogy, an object of class “convertible” certainly *is an* object of the more *general* class “automobile,” but more *specifically*, the roof can be raised or lowered.



1.15 Keeping Up-to-Date with Information Technologies

- ▶ Figure 1.17 lists key technical and business publications that will help you stay up-to-date with the latest news and trends in computer, Internet and web technology.



Publication	URL
ACM TechNews	technews.acm.org/
ACM Transactions on Accessible Computing	www.is.umbc.edu/taccess/index.html
ACM Transactions on Internet Technology	toit.acm.org/
Bloomberg BusinessWeek	www.businessweek.com
CNET	news.cnet.com
Communications of the ACM	cacm.acm.org/
Computer World	www.computerworld.com
Engadget	www.engadget.com
eWeek	www.ewEEK.com
Fast Company	www.fastcompany.com/
Fortune	money.cnn.com/magazines/fortune/
IEEE Computer	www.computer.org/portal/web/computer

Fig. 1.17 | Technical and business publications. (Part 1 of 2.)



Publication	URL
IEEE Internet Computing	www.computer.org/portal/web/internet/home
InfoWorld	www.infoworld.com
Mashable	mashable.com
PCWorld	www.pcworld.com
SD Times	www.sdtimes.com
Slashdot	slashdot.org/
Smarter Technology	www.smartertechnology.com
Technology Review	technologyreview.com
Techcrunch	techcrunch.com
Wired	www.wired.com

Fig. 1.17 | Technical and business publications. (Part 2 of 2.)