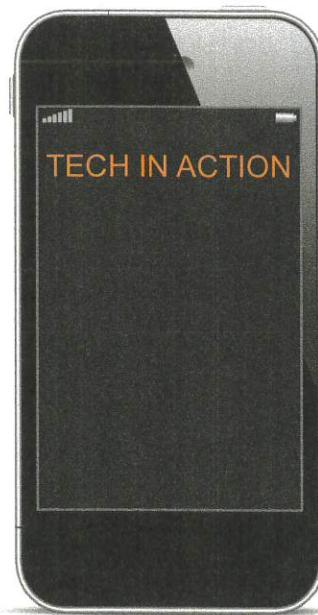


Step 6 Move to the Corona Simulator window and click **File->Open Project**. Then navigate to **main.lua** and open it.

You have just written a complete iPhone application! You can also run it on an iPad or any Android phone or tablet. Click **View->View As** and see how the program output looks on a few different devices.

Step 7 Play with changing the color by altering the values in `setTextColor`. Try adding the line `myText.rotation = 45`.

Visit docs.coronalabs.com/api to see the full set of built-in methods available. Look over the physics library and begin to imagine what you could create! (Courtesy of Corona Labs, Inc.)



(LOVE graphic/Shutterstock)

Make This



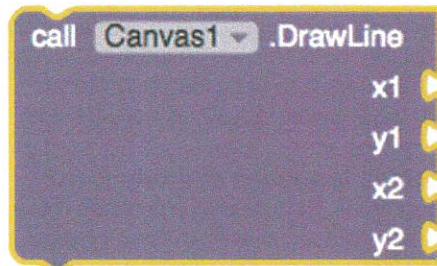
TOOL: AI Canvas, If Block

MAKE: A Notepad

Drawing and writing on a touch device is convenient and creative.

In this exercise, you'll continue your mobile app development by using the Canvas component in App Inventor to build a notepad that lets you switch pen colors and draw shapes.

The Canvas component lets you draw on the screen. Combine that with statements like "if" and "for" and you can make more sophisticated behaviors.



Use the Canvas component in App Inventor to combine control statements with drawing commands to create your own images.

For the instructions for this exercise, please go to pearsonhighered.com/techinaction or MyITLab.

Programming Languages

Earlier in the chapter, you learned about the five main categories (generations) of programming languages. In this section, we discuss the specific programming languages that are members of these different generations.



many languages for MANY PROJECTS

In any programming endeavor, programmers want to create a solution that meets several objectives. The software needs to:

- Run quickly
- Be reliable
- Be simple to expand later when the demands on the system change
- Be completed on time
- Be finished for the minimum possible cost

Because it's difficult to balance these conflicting goals, many programming languages have been developed. Although programming languages often share common characteristics, each language has specific traits that allow it to be the best fit for certain types of projects. The ability to understand enough about each language to match it to the appropriate style of problem is an exceptionally powerful skill for programmers.

What languages are popular today? One quick way to determine which languages are popular is to examine job postings for programmers. As of this writing, the languages most in demand include C/C++ and Java. In specific industries, certain languages tend to dominate the work. In the banking and insurance industries, for example, the programming language COBOL is still common, although most other industries rarely use it anymore. The TIOBE Index, shown in Figure 10.20, uses a number of different techniques to get a feel for which languages are popular in the software industry at certain times.

How do I know which language to study first? A good introductory programming course will emphasize many skills and techniques that will carry over from one language to another. You should find a course that emphasizes design, algorithm development, debugging techniques, and project management. All of these aspects of programming will help you in any language environment.

Visual languages like App Inventor or Scratch are often used to introduce key programming concepts. Many colleges and universities have opted to have students begin with Java, C++, or Python. Students interested in the arts are often using the language **Processing** as a first experience with coding.

How does anyone learn so many languages?

Professional programmers can become proficient at new languages because they've become familiar with the basic components, discussed in this chapter's Dig Deeper feature, that are common to all languages: syntax, keywords, operators, and data types.

Sep 2014	Sep 2013	Change	Programming Language	Change
1	1		C	-0.25%
2	2		Java	-2.01%
3	4	▲	Objective-C	+1.37%
4	3	▼	C++	-3.99%
5	6	▲	C#	-1.21%

FIGURE 10.20 When attacking a problem, you can choose from many different programming languages. This chart of the TIOBE Index shows how popular certain languages are at specific times. (*Courtesy of TIOBE Software*)



SOUND BYTE

Programming with the Processing Language

In this Sound Byte, you'll use the Processing language to create interactive art.

Selecting the Right Language

How do programmers know which language to select for a specific project? A programming team considers several factors before selecting the language it will use for a specific project:

- **Space available:** Not all languages produce code that takes up the same amount of space. Therefore, the target language should be well matched to the amount of space available for the final program. For example, if the program will be embedded in a chip for use in a cell phone, it's important for the language to create space-efficient programs.
- **Speed required:** Although poorly written code executes inefficiently in any language, some languages can execute more quickly than others. Some projects require a focus on speed rather than size. These projects require a language that produces code that executes in the fastest possible time.
- **Organizational resources available:** Another consideration is the resources available in a manager's group or organization. Selecting a language that's easy to use and that will be easy to maintain if there's a turnover in programmers is an important consideration.
- **Type of target application:** Certain languages are customized to support a specific environment (UNIX or Windows,

BITS&BYTES

Coding for Zombies

There are an incredible number of free courses available for the latest programming technologies. Rails for Zombies from Code School is one such course. The free interactive coding environment runs right in your browser and teaches you to create a Twitter-like program for Zombies using the Ruby on Rails web application framework. Also, explore course providers like Coursera, Udacity, and codecademy. They offer courses complete with quizzes, video lectures, and live discussion on topics like Python, Web Programming, and Java.

for instance). Knowing which languages are most commonly used for which environments can be helpful. We'll discuss this more in the next section.

Can I just point and click to create an application?

Many languages have a development environment that features a drag and drop-style interface for designing the visual layout of an application. Programmers use a mouse to lay out the screen and position scroll bars and buttons. The code needed to explain this to the computer is then written automatically. **Visual programming languages**, like Scratch and App Inventor, go even further. They use graphical blocks to represent control elements and variables. Programming consists of clicking together these blocks to define program behavior. ■

In the field of software programming, mistakes can be costly. Consider these examples:

- In 1996, Europe's newest unmanned rocket, the Ariane 5, had to be destroyed seconds after launch, along with its cargo of four scientific satellites. Cost: \$500 million. The problem occurred when the guidance computer tried to convert rocket velocity from a 64-bit to a 16-bit value. Programmers had not properly coded for this, and an overflow error resulted. When the guidance system shut down, control passed to the backup unit, which also failed because it was running the same algorithm.
- When one switch in the AT&T network had a minor problem in 1990, system software fired off a message to the other 113 AT&T switching centers. This sparked a cascade of shutdowns, bringing down the entire network for nine hours. An estimated 75 million phone calls were missed, and over 200,000 airline reservations were lost. A single line of code in a large software upgrade was in error and led to the shutdown.
- In 2007, Los Angeles International Airport grounded 17,000 planes. For over eight hours, no one was authorized to enter or leave the United States through LAX. Software on a network card was sending faulty data and brought the entire Customs and Border Protection network to a halt.

Could better software engineering practices have prevented these failures? Many different factors contributed to this set of accidents, including the following:

- Simple programming errors
- Inadequate safety engineering
- Poor human-computer interaction design
- Inadequate or inappropriate testing
- Too little focus on safety by the manufacturing organization
- Inadequate reporting structure at the company level

Who should be held responsible for defective software? Is it the corporate management, who did not institute a defined software process? Is it

the production managers, who forced tight schedules that demanded risky software engineering practices? What about the software engineers who wrote the defective code? What about the users of the software? Can they be held responsible for accidents? What if they made changes to the system?

Organizations of engineers and software designers have tried to define ethical standards that will minimize the risk of negative impact from their work. The very first article of the code of ethics of the Institute of Electrical and Electronics Engineers (IEEE) alerts engineers to their responsibility for making decisions that protect the health and safety of others. The Association for Computing Machinery (ACM) and the IEEE have established eight principles for ethical software engineering practices:¹

1. *Public:* Software engineers shall act consistently with the public interest.
2. *Client and Employer:* Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.
3. *Product:* Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
4. *Judgment:* Software engineers shall maintain integrity and independence in their professional judgment.
5. *Management:* Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
6. *Profession:* Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
7. *Colleagues:* Software engineers shall be fair to and supportive of their colleagues.
8. *Self:* Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Can you think of other steps the software industry could take to make sure its work contributes positively to our society?

¹"Software Engineering Code of Ethics and Professional Practice," by IEEE-CS/ACM Joint Task Force on Software Engineering Ethics and Professional Practices. Copyright © 2011 by the Institute of Electrical and Electronics Engineers. Reprinted by permission.



exploring programming LANGUAGES

All programming languages have many features in common. Let's take a tour of many of the popular languages and learn what makes each one special for particular situations.

Visual Basic

Why do programmers choose Visual Basic?

Programmers often like to build a **prototype**, or small model, of their program at the beginning of a large project. Prototyping is a form of **rapid application development (RAD)**, an alternative to the waterfall approach of systems development that was described at the beginning of this chapter. Instead of developing detailed system documents before they produce the system, developers create a prototype first, then generate system documents as they use and remodel the product.

Prototypes for Windows applications are often coded in Microsoft **Visual Basic (VB)**, a powerful programming language used to build a wide range of Windows applications. One strength of VB is its simple, quick interface, which is easy for a programmer to learn and use. It has grown from its roots in the language BASIC (short for Beginner's All-purpose Symbolic Instruction Code) to become a sophisticated and full-featured object-oriented language.

VB is designed for building object-oriented applications for Windows, the web, and mobile devices. Figure 10.21 shows how the interface of VB makes it easy to drag and drop entire programming components into an application. VB and the .NET Framework are both part of Visual Studio, which provides a complete set of developer tools.

How does the Microsoft .NET Framework help programmers?

The Microsoft .NET (pronounced "dot net") Framework is a software development environment designed to let websites "talk" to each other easily. The .NET Framework includes **web services**, programs that a website uses to make information available to other websites. Web services provide a standard way for software to interact. For example, a web application could use the Google web service to search for information or to check the spelling of a word. The Google web service returns the requested information to your program in a standard format.

C and C++

What languages do programmers use if the problem requires a lot of "number crunching"? A Windows application

that demands raw processing power to execute difficult, repetitive numerical calculations is most often a candidate for C/C++. For example, applications that simulate human cells and drug interactions have to solve elaborate mathematical equations many thousands of times each second and are, therefore, excellent candidates for programming using C/C++. Several companies sell C/C++ design tools equipped with an environment that makes Windows programming as visual as with VB.

Why was the C language developed? The predecessor of C++, **C**, was originally developed for system programmers. It was defined by Brian Kernighan and Dennis Ritchie of AT&T Bell Laboratories in 1978 as a language that would make accessing the operating system easier. It provides higher-level programming language features (such as *if* statements and *for* loops) but still allows programmers to manipulate the system memory and CPU registers directly. This mix of high- and low-level access makes C highly attractive to "power" programmers. Most modern operating systems (Windows, OS X, and Linux) were written in C.

The **C++** language takes C to an object-oriented level.

Bjarne Stroustrup, the developer of C++, used all the same symbols and keywords as C, but he extended the language with additional keywords, better security, and more support for the reuse of existing code through object-oriented design.

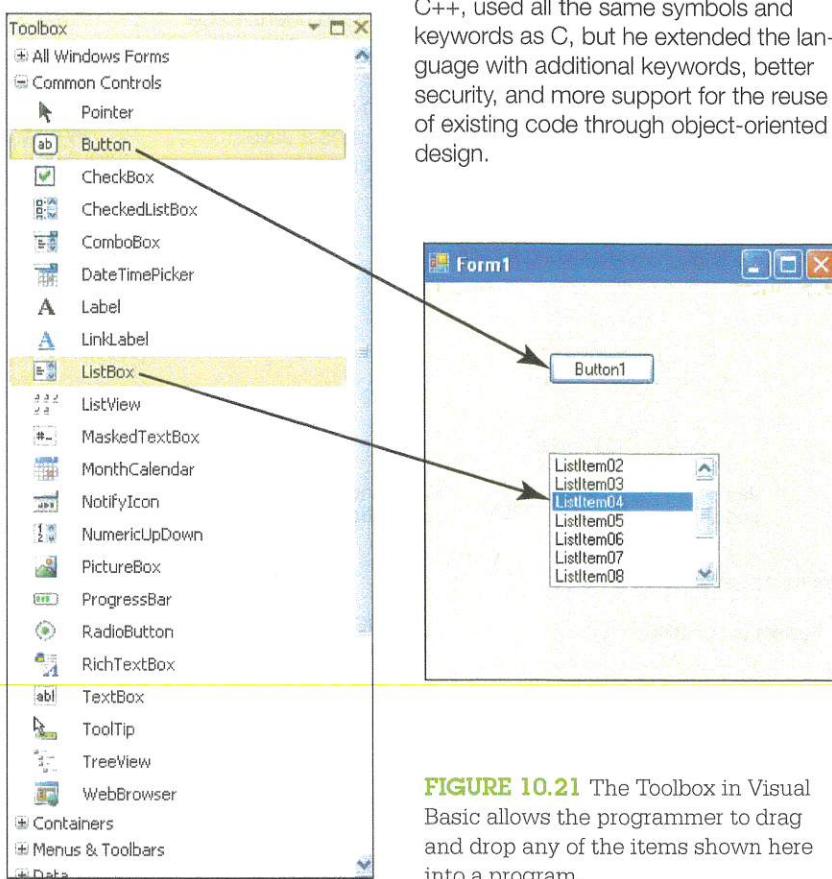


FIGURE 10.21 The Toolbox in Visual Basic allows the programmer to drag and drop any of the items shown here into a program.

Are C and C++ natural choices for when I'm looking to learn my first language? Neither C nor C++ was intended as a teaching language. The notation and compactness of these languages make them relatively difficult to master. They're in demand in industry, however, because C/C++ can produce fast-running code that uses a small amount of memory. Programmers often choose to learn C/C++ because their basic components (operators, data types, and keywords) are common to many other languages.

Java and C#

What language do programmers use for applications that need to collect information from networked computers? Java would be a good choice for these types of applications. James Gosling of Sun Microsystems introduced Java in the early 1990s. It quickly became popular because its object-oriented model enables Java programmers to benefit from its large set of existing classes. For example, a Java programmer could begin to use the existing "network connection" class with little attention to the details of how that code itself was implemented. Classes exist for many graphical objects, such as windows and scroll bars, and for network objects such as connections to remote machines. Observing Java's success, Microsoft released a language named **C#** (pronounced "see sharp") that competes with Java.

Can a Java application work on any type of computer? An attractive feature of Java is that it's **architecture neutral**. This means that Java code needs to be compiled only once, after which it can run on many CPUs (see Figure 10.22). The Java program doesn't care which CPU, OS, or user interface is running on the machine on which it lands. This is possible because the target computer runs a Java Virtual Machine (VM), software that can explain to the Java

program how to function on any specific system. A Java VM installed with Microsoft Internet Explorer, for example, allows Internet Explorer to execute any **Java applet** (small Java-based program) it encounters on the Internet. Although Java code doesn't perform as fast as C++, the advantage of needing to compile only once before a program can be distributed to any system is extremely important.

Objective C and Swift

What's the most popular language for writing OS X applications? Objective C is the language most often used to program applications to run under OS X. It's an object-oriented style of language, a superset of the C language, so it includes all the keywords of C and then adds more keywords and features. It's often used together with a library called Cocoa. The Cocoa library, or framework, lets users program for the OS X graphical user interface. The Cocoa Touch extension introduces a framework of methods that support gesture recognition for touch devices like the iPhone or iPad.

Swift is a new programming language introduced by Apple for developing for iOS and OS X. Easier to read and maintain, Swift uses a "playground" area where you can quickly write and test ideas before moving the code into the main project.

Is there a favorite IDE for Objective C? Many different IDE tools support Objective C, but OS X ships with a tool named Xcode (see Figure 10.23) that's often used to develop Objective C applications for OS X. Xcode is available free from the online Mac App Store.

HTML

What's the most basic formatting language for web applications? A document that will be presented on

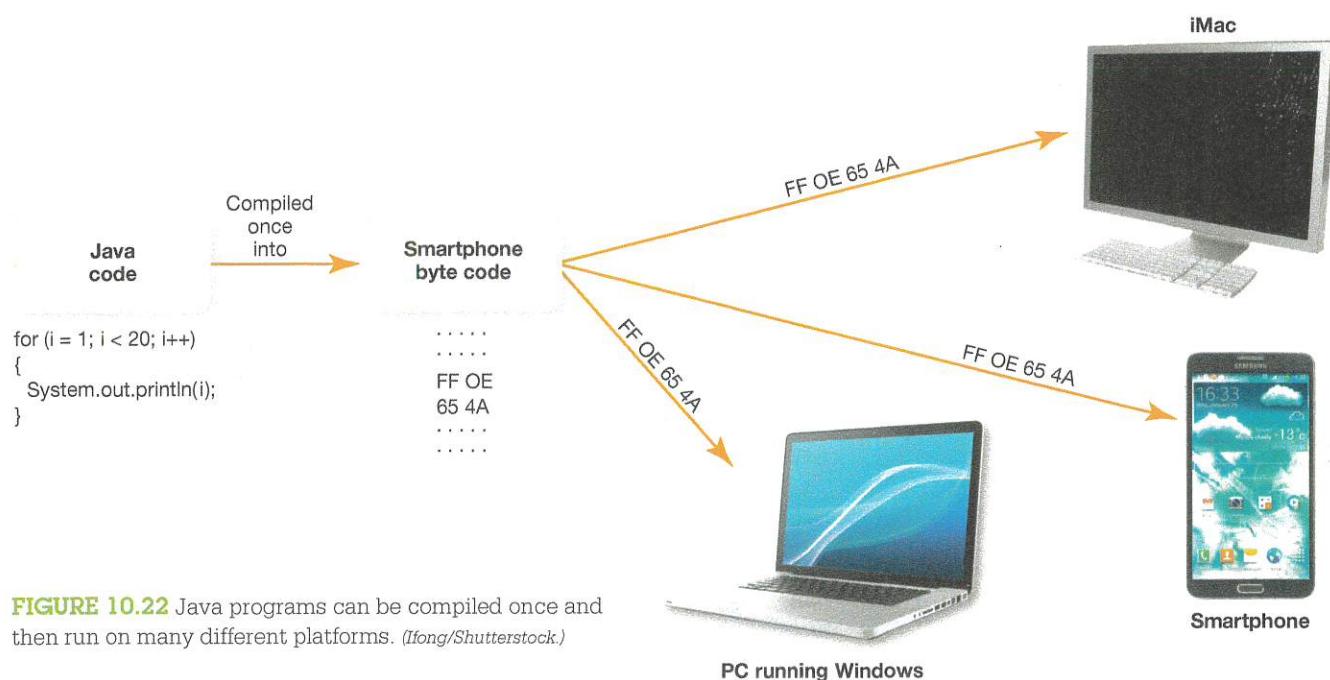


FIGURE 10.22 Java programs can be compiled once and then run on many different platforms. (*Ifong/Shutterstock*)

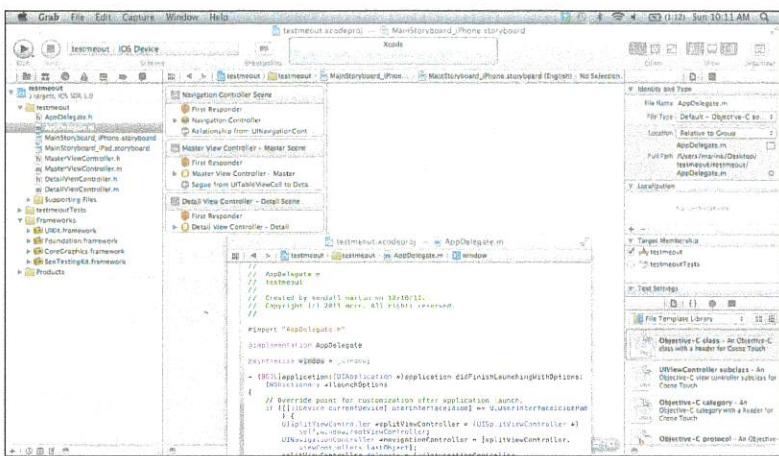


FIGURE 10.23 The Xcode IDE tool provides a supportive environment for developing OS X applications. (Screen shot(s) reprinted with permission from Apple, Inc.)

the web must be written using special symbols called **tags**. Tags control how a web browser will display the text, images, and other content tagged in **Hypertext Markup Language (HTML)**.

Although knowledge of HTML is required to program for the web, HTML itself isn't a programming language. HTML is just a series of tags that modify the display of text. HTML was the original standard defining these tags. Many good HTML tutorials are available on the web at sites such as Learn the Net (learnthenet.com), Webmonkey (webmonkey.com), and the World Wide Web Consortium (w3.org). These sites include lists of the major tags that can be used to create HTML and XHTML documents.

Are there tools that help programmers write in HTML?

Several different programs are available to assist in the generation of HTML. Adobe Dreamweaver and Microsoft Expression Web present web page designers with an interface that's similar to a word processing program. Web designers can quickly insert text, images, and hyperlinks, as shown in Figure 10.24. The program automatically inserts the corresponding HTML tags. For simple, static web pages, no programming is required.

Will HTML continue evolving? Yes, the current release of HTML is HTML5. HTML standards are discussed and finalized by committee, and HTML5 is a significant advancement. For example <audio> and <video> tags allow pages to easily integrate media and let designers more easily support manipulating those elements. HTML5 eliminates the need to install third-party browser plug-ins and supports features like drag-and-drop elements and document editing.

JavaScript and VBScript

Which programming languages do programmers use to make complex web pages? To make their web pages more visually

appealing and interactive, programmers use scripting languages to add more power and flexibility to their HTML code. A **scripting language** is a simple programming language that's limited to performing a set of specialized tasks. Scripts allow decisions to be made and calculations to be performed. Several popular scripting languages work well with HTML, including JavaScript, PHP (Hypertext Preprocessor), and VBScript.

JavaScript is a scripting language that's customized to work with the elements of a web page and is often used to add interactivity to web pages. The syntax, keywords, data types, and operators are subsets of Java. In addition, JavaScript has a set of classes that represent the objects often used on web pages: buttons, check boxes, and drop-down lists.

The JavaScript button class, for example, describes a button with a name and a type—for example, whether it's a regular button or a Submit or Reset button. The language includes behaviors, such as click(), and can respond to user actions. For example, when a user moves his or her mouse over a button and clicks to select it, the button "knows" the user is there and jumps in and performs a special action (such as playing a sound).

Are there other scripting languages besides JavaScript?

JavaScript? Programmers who are more familiar with Visual Basic than Java or C++ often use **VBScript**, a subset of Visual Basic, to introduce dynamic decision making into web pages. **Dynamic decision making** means that the page can decide how to display itself based on the choices the reader makes. PHP, discussed in the following section, is another scripting language that has become extremely popular. It's a free, open-source product that runs very efficiently on multiple platforms, including Windows, UNIX, and Linux.

ASP, JSP, and PHP

How are interactive web pages built? To build websites with interactive capabilities, programmers use **Active Server Pages (ASP)**, **JavaServer Pages (JSP)**, or the scripting language **PHP (Hypertext Preprocessor)**.

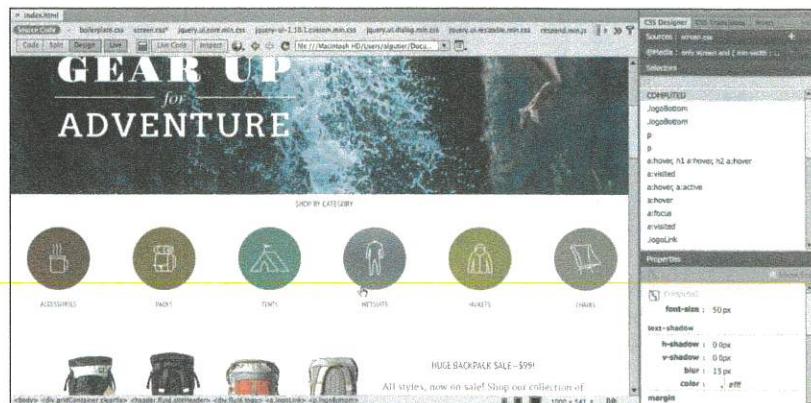


FIGURE 10.24 Adobe Dreamweaver is a popular tool for creating web pages. (Adobe Systems, Inc.)

BITS & BYTES

The Best Résumé

Ever heard of GitHub? It's an online site for storing and managing software. One aspect of GitHub is that you can easily find projects or people who excite you and follow their work. For example, you can follow a specific person and get notifications of his or her work or even watch a specific project. When the project or work of a particular person is updated, you'll see that right away in your GitHub dashboard. You can also establish teams or "organizations" and work with multiple developers. Fork a copy of the code for a project you admire, and when you add a new feature, send out a request to have it pulled back into the original code.

And what's really cool is that your GitHub activity becomes a brilliant component of your résumé. A graph documents the contributions you have made over the previous year and allows an employer to actually see and use your body of work. A solid portfolio of your coding projects in GitHub is a smart investment in marketing yourself down the road.

to adapt the HTML page to the user's selections. The user supplies information that's translated into a request by the main computer at the company that owns the website, often using a database query language such as SQL. Scripting code in ASP, JSP, or PHP controls the automatic writing of the custom HTML page that's returned to the user's computer.

What does additional programming bring to a web page? The most advanced web pages interact with the user, collecting information and then customizing the content displayed based on the user's feedback. For example, the client/server type application shown in Figure 10.25 shows the web page of the ABC Bike Company collecting a customer bicycle inquiry for red bikes. The program then asks ABC's main server for a list of red bicycles sold by the company. A PHP program running on the server creates a new HTML page and delivers that to the user's browser, telling the customer what red bikes (including details such as model and size) are currently sold by ABC.

Server programs, written in ASP or PHP, can have HTML code as their output. They use what the user has told them (via the list boxes, check boxes, and buttons on the page) to make decisions. Based on those results, the server software decides what HTML to write. A small example of ASP writing its own HTML code is shown in Figure 10.26.

AJAX, XML, and JSON

What if a programmer wants to create a web application that smoothly updates and communicates with other computers? Many websites feature elaborate animations that interact with visitors. They may reach out to several other web resources to gather information or request services like translation. The collection of technologies referred to as **AJAX (Asynchronous JavaScript and XML)** and the continued evolution of HTML5 allow such web applications to update information on a page without requiring the user to do a page refresh or leave the page. By using existing technologies to do more processing in the browser, users have a more responsive experience.

How do programs gather information from other computers? The markup language called

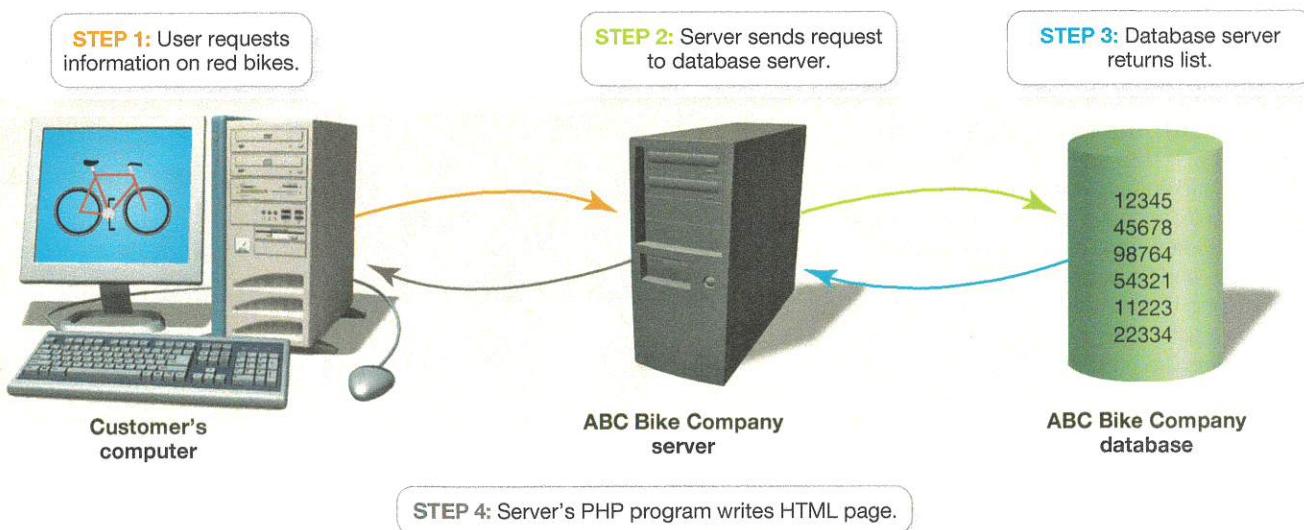


FIGURE 10.25 An online store is an example of the client/server type of Internet application.

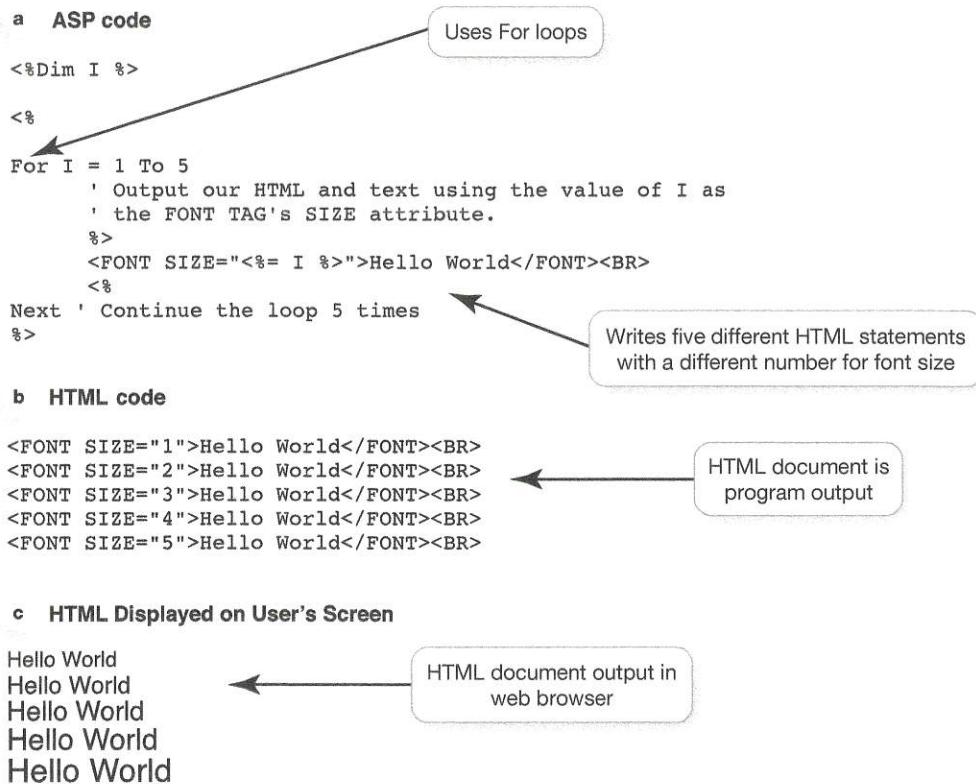


FIGURE 10.26 An ASP program can (a) write HTML code as its (b) output. (c) This image illustrates how the HTML page would show up in a browser.

eXtensible Markup Language (XML) enables designers to define their own data-based tags, making it much easier for a program running on a server computer (a web service) to transfer the key information on its page to another site. When websites communicate with humans, HTML works well because the formatting it controls is important. People respond immediately to the visual styling of textual information; its layout, color, size, and font design all help to transfer the message of the page to the reader. When computer programs want to communicate with each other, however, all of these qualities just get in the way.

With XML, groups can agree on standard systems of tags that represent important data elements. For example, the XML tags `<stock>` and `</stock>` might delimit key stock quote information. Mathematicians have created a standardized set of XML tags named MathML for their work, and biometrics groups continue to refine XML standards to describe and exchange data such as DNA and face scans. Without XML, a website that wanted to look up current stock pricing information at another site would have to retrieve the HTML page, sort through the formatting information, and try to recognize which text on the page identified the data needed. **JSON** (short for JavaScript Object Notation) is another standard for exchanging information between a server computer process and a client.

The information is delivered as a series of names and their values, making it easy for programs to parse and easy for humans to read.

Figure 10.27 shows a table of popular programming languages and technologies with their features and the typical settings in which they're used.

Mobile Applications

How do programmers build applications for mobile devices?

Special languages and supporting tools are speeding the development of applications for mobile devices like smartphones and tablets. Programmers need to be able to take advantage of specific features like GPS capability, compasses, software keyboards, and touch-sensitive screens. In addition, the user interface has to take the smaller screen size of mobile devices into account.

What development tools are used for creating mobile apps for Apple's iOS platform?

To start a complex project like an iPhone app requires a detailed prototype. Each of the many screens, all the user interface elements, and all the content need to be organized and linked smoothly. Often, programmers begin with a prototype,

FIGURE 10.27

Popular Programming Languages		
PROGRAMMING LANGUAGE	FEATURES	TYPICAL SETTING
C/C++ and C#	<ul style="list-style-type: none">Can create compact code that executes quicklyProvides high- and low-level access	<ul style="list-style-type: none">Used in industrial applications such as banking and engineering
Java	<ul style="list-style-type: none">Is architecture neutralIs object oriented	<ul style="list-style-type: none">Used to create applets that can be delivered over the web
Objective C	<ul style="list-style-type: none">Has a framework for writing iOS applications	<ul style="list-style-type: none">Used to create applications for OS X and Apple mobile devices
Visual Basic	<ul style="list-style-type: none">Is easy to learn and useIs object orientedHas a drag-and-drop interface	<ul style="list-style-type: none">Used in prototype developmentUsed to design graphical user interfaces
WEB TECHNOLOGIES	FEATURES	TYPICAL SETTING
AJAX	<ul style="list-style-type: none">Uses a combination of existing technologies like JavaScript, CSS, and XML	<ul style="list-style-type: none">Creates websites that can update without the user refreshing the page
HTML5	<ul style="list-style-type: none">Latest version of HTML	<ul style="list-style-type: none">Introduces tags like <video> and supports drag and drop
VBScript	<ul style="list-style-type: none">Is similar in syntax to Visual BasicHas classes that represent buttons, drop-down lists, and other web page components	<ul style="list-style-type: none">Creates code that lives on the client machine and adds interaction to web pages
XML	<ul style="list-style-type: none">Enables users to define their own tags	<ul style="list-style-type: none">Facilitates exchange of information from web services
JSON	<ul style="list-style-type: none">Format defined with name/value pairs	<ul style="list-style-type: none">Very common format for exchange of information from web services

created quickly with drag-and-drop elements using products like Proto.io or Interface Builder. Proto.io (<https://proto.io>) is a browser-based tool that lets you construct a working simulation of your application. Interface Builder is part of the Apple Xcode development tool; it requires a bit more expertise to use than Proto.io but can also rapidly create a prototype.

When it's time to begin writing the code for an iOS app, programmers turn to Objective C and use the Apple Xcode development toolset. Xcode lets designers code and debug the behavior of the application and simulate the application in a software version of the target device. After the program is running, its performance can be profiled for speed, memory usage, and other possible problems.

Are there different tools for building apps for Android devices? Yes, the Android software development kit (SDK) is required to build apps targeting Android smartphones and tablets. There are many ways programmers work with the Android SDK, including using well-known IDEs like Eclipse with special plug-ins or using the Android Studio, shown in Figure 10.28. Information on the latest version of the Android SDK, as well as tutorials, guides, and other resources, is available at the Android Developers page (developer.android.com).

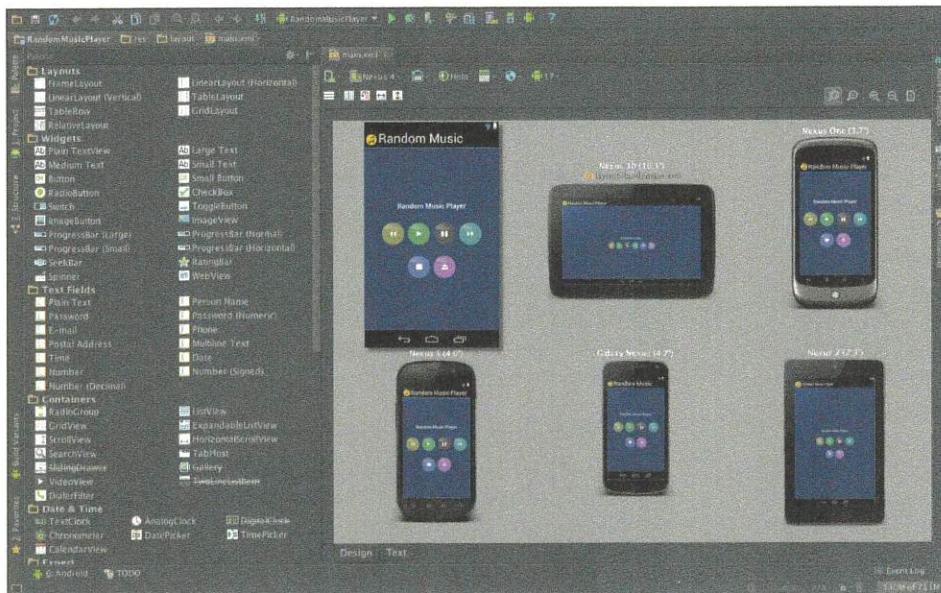


FIGURE 10.28 Android Studio integrates all the tools required to program Android applications from scratch. (Courtesy of Google, Inc.)

trends in IT

Unity is a game-development environment that supports home video game production by combining many tools into one package, giving you what you need to edit, test, and play your game idea quickly. It can generate real-time shadowing effects, for example, and has preprogrammed scripts integrated. After you've polished your production, Unity allows you to publish for web, mobile, or console platforms. With over 500,000 developers using Unity, and over 60 million installs of the Unity browser plug-in, the potential of this tool is amazing (see Figure 10.29).

You'll start with the built-in Unity editor so you can quickly assemble a 3-D environment. Unity gives you access to its Asset Store right from the editor window. Art assets are supplied so that you instantly have materials to create your own landscape. Tutorials and sample projects are available, as well as character models and textures. Grab a sound effect or some code to simulate a great explosion effect. If you create assets you'd like to distribute, you can put them up for sale within the Asset Store.

Unity incorporates lighting control and shading and the application of textures in order to develop



FIGURE 10.29 Unity is a free game engine that can build web, mobile, or console games. (*Fanatic Studio/Getty Images*)

Emerging Technologies: Unite All Your Video Game Design Tools

rich terrains. Its built-in physics engine lets objects respond to gravity, friction, and collisions. One example is the “wheel collider,” a module that simulates the traction of real car tires. Audio processing is included, with effects like echo, reverb, and a chorus filter, as are tools to make real-time networking available from your game.

Programming is handled through several flexible scripting languages—JavaScript, C#, and an implementation of Python named Boo. It has a visual interface that lets you drag and drop objects to set variables, or you can choose to sync with an external IDE like Visual Studio. There's an integrated debugger so that you can pause your game, check variable values, and quickly repair errors. When it's time to optimize the performance before your final release, you can use the built-in profiler: It reports statistics on where the processing time is being spent so that you can fine tune game play.

Unity is available in a free downloadable version (unity3d.com) and in a pay version, Unity 3D Pro. The Pro version adds features but also can be licensed and used by companies. The free version of Unity can be used to create games that can be sold—there is no royalty or revenue sharing required. And there are modules that let you easily port your code to a number of target platforms: PC, iOS, Android, and web as well as Xbox 360, PS4, and Wii.

You'll also find an active community to support you as you learn. There are live online training courses offered and forums for community discussion. There is also a dedicated IRC (Internet Relay Chat) channel for live chatting with Unity users. If you need help distributing your finished game, you can join Union, a service that brings your game to new markets for you.

If you're interested in video game development, architectural visualizations, or just creating interactive animations, download Unity and realize the power of programming.

I'm not a programmer, but can I make a simple app? Absolutely. Tools like Corona and App Inventor produce amazing games and apps quickly (see the Try This on pages 428–429 for more information). Tens of millions of apps constructed using Corona have been downloaded, including the overnight sensation Bubble Ball created by 14-year-old Robert Nay (naygames.com). The apps you have written using App Inventor in the Make This exercises can be uploaded to the Google Play store for download worldwide.

If your goal is to make a mobile app that's very simple or even specific to one occasion, like a wedding, there are web-based products that make that quick and easy. Magmito (magmito.com) supports developing a simple app with text and graphics and requires no programming knowledge.

Does an application need to be rewritten for every kind of mobile device?

Programming environments like Corona and Magmito support publishing an application to several different types of devices. Although these tools can be great time savers for very simple applications, for programmers using specific features that make that device unique or for those concerned with extracting ultimate performance, custom programming for each environment still is required.

The Next Great Language

What will the next great language be? It is never easy to predict which language will become the next "great" language. Software experts predict that as software projects continue to grow in size, the amount of time needed to compile a completed project will also grow. It's not uncommon for a large project to require 30 minutes or more to compile. Interpreted languages, however, take virtually no compile time because compilation occurs while the code is being edited. As projects get larger, the capability to be compiled instantaneously will become even more important. Thus, interpreted languages such as Python, Ruby, and Smalltalk could become more important in the coming years.

Will all languages someday converge into one?

Certain characteristics of modern programming languages correspond well with how programmers actually think. These traits support good programming practices and are emerging as common features of most modern programming languages. The object-oriented paradigm is one example. Both Visual Basic and COBOL have moved toward supporting objects.



FIGURE 10.30 There will always be a variety of languages, each with its own personality. (Luis Guillermo Restrepo Rivas)

There will always be a variety of programming languages, however. Figure 10.30, in which the artist tries to give each language a "personality," illustrates that idea in a lighthearted fashion.

Forcing a language to be so general that it can work for any task also forces it to include components that make it slower to compile, produce larger final executables, and require more memory to run. Having a variety of languages and mapping a problem to the best language create the most efficient software solutions.

So what do I do if I want to learn languages that will be relevant in the future? No particular set of languages is best to learn, and there is no one best sequence in which to learn them. The Association for Computing Machinery (acm.org) encourages educators to teach a core set of mathematical and programming skills and concepts, but school and university departments are free to offer a variety of languages.

When you're selecting which programming languages to study, some geographical and industry-related considerations come into play. For example, in an area in which a large number of pharmaceutical companies exist, there may be a demand for Massachusetts General Hospital Utility Multi-Programming System (MUMPS). This language is often used to build clinical databases, an important task in the pharmaceutical industry. Review the advertisements for programmers in area newspapers and investigate resources such as ComputerJobs (computerjobs.com) to identify languages in demand in your area.

Regardless of whether you pursue a career in programming, having an understanding of how software is created will help you in many IT careers. Software is the set of instructions that allows us to make use of our hardware. Programming skills give you the power to understand, create, and customize a computer system. ■

SOUND BYTE

Using the Arduino Microcontroller

In this Sound Byte, you'll use the Arduino microcontroller to control LED lights and motors.

Before moving on to the Chapter Review:

- Watch Replay Video 10.2 .
- Then check your understanding of what you've learned so far.

check your understanding // review & practice

For a quick review to see what you've learned so far, answer the following questions. Visit pearsonhighered.com/technaction to check your answers.

multiple choice

1. Selecting the right programming language for a project depends on
 - a. who you have working for you and what they know.
 - b. the final requirements for performance and space used.
 - c. the environment of the finished project.
 - d. all of the above.
2. An IDE is a tool that facilitates
 - a. editing.
 - b. debugging.
 - c. compiling.
 - d. all of the above.
3. Which is NOT an advantage of Java?
 - a. Java is architecture neutral.
 - b. Java needs to compile only once prior to distribution.
 - c. Java supports network communications.
 - d. Java performs faster than C++.
4. Which is TRUE about XML?
 - a. XML supports the development of rich multimedia.
 - b. XML makes it possible to update web pages without refreshing.
 - c. XML enables designers to define their own data-based tags.
 - d. XML has classes that represent drop-down lists and other web elements.
5. HTML5 features helpful new tags like
 - a. <audio> and <video>
 - b. <movie> and <actor>
 - c. <metaheader>
 - d. <div>

To take an autograded version of this review, please go to the companion website at pearsonhighered.com/technaction, or go to your MyITLab course.

Continue 

10 Chapter Review

summary //



The Importance of Programming

1. Why do I need to understand how to create software?

- Programming skills allow you to customize existing software products to accomplish required tasks.
- A beginning-level knowledge of programming will let you create macros, customized mini-programs that speed up redundant tasks.

sequence of steps that the program must take to complete the work.

- The algorithm is then translated into highly structured programming code.
- The code goes through the processes of debugging, in which the programmers find and repair any errors in the code.
- Testing is performed by the programming team and by the people who will use the program.
- The results of the entire project are documented for the users and the development team.



The Life Cycle of an Information System

2. What is a system development life cycle, and what are the phases in the cycle?

- An information system includes data, people, procedures, hardware, and software.
- The set of steps followed to ensure that development proceeds in an orderly fashion is the system development life cycle (SDLC). There are six steps in the SDLC waterfall model:
 1. A problem or opportunity is identified.
 2. The problem is analyzed, and a program specification document is created to outline the project objectives.
 3. A detailed plan for programmers to follow is designed using flowcharts and data-flow diagrams.
 4. Using the developed plan, programmers develop the program, and the program is then documented.
 5. The program is tested to ensure that it works and that it's installed properly.
 6. Ongoing maintenance and evaluation ensure a working product.

4. What role does a problem statement play in programming?

- The problem statement is an explicit description of what tasks the computer program must accomplish and how the program will execute these tasks and respond to unusual situations.
- The problem statement describes the input data that users will have at the start of the job, the output that the program will produce, and the exact processing that converts these inputs to outputs.
- The problem statement identifies potential errors and plans to address these errors.

5. How do programmers create algorithms and move from algorithm to code?

- Programmers create an algorithm by converting a problem statement into a list of steps and identifying where decision points occur.
- Yes/no binary decisions are common, and often the pattern of a repeating action loop is recognized.
- Algorithms are documented in the form of a flowchart or in pseudocode.
- Programmers use either top-down or object-oriented analysis to produce the algorithm.
- Computer code uses special words and strict rules to enable programmers to control the CPU without having to know all of its hardware details.
- Programming languages are classified in several major groupings, sometimes referred to as *generations*, with the first generation being machine language—the binary code of 1s and 0s that the computer understands. Assembly language is the next generation; it uses short, English-like commands that speak directly to the CPU and give the programmer direct control of hardware.



The Life Cycle of a Program

3. What is the life cycle of a program?

- The problem statement identifies the task to be computerized and describes how the software program will behave.
- An algorithm is developed that specifies the

resources. Each successive generation in language development has relieved programmers of some of the burden of keeping track of what the hardware requires and more closely matches how humans think about problems.

- Compilation is the process by which code is converted into machine language, the language the CPU can understand. A compiler is a program that understands both the syntax of the programming language and the exact structure of the CPU and its machine language. It can translate the instructions written by programmers in the higher-level language into machine language, the binary patterns that will execute commands on the CPU.
- Each programming language has its own compiler. Separate versions are required to compile code that will run on each different type of processor.

6. What steps are involved in completing the program?

- If programmers make errors in the algorithm or in translating the algorithm to code, problems will occur. Programmers debug the program by running it constantly to find errors and to make sure the program behaves the way it should.
- Once debugging has detected all the code errors, users—both within the company and outside the company—test the program in every way they can imagine, both as the program was intended to be used and in ways only new users might think up.
- Before its commercial release, software is often provided at a reduced cost or at no cost in a beta version to certain test sites or to interested users for a last round of testing.
- Once testing is complete, technical writers create internal documentation for the program and external documentation that will be provided to users of the program. User training, which begins once the software is distributed, teaches the user community how to use the software efficiently.



Many Languages for Many Projects

7. How do programmers select the right programming language for a specific task?

- A programming team reviews several considerations before selecting the language to be used. Certain languages are best used for certain problems.

- The target language should be well matched to the amount of space available for the final program.
- Some projects require the selection of a language that can produce code that executes in the fastest possible time.
- Selecting a language with which the programmers are familiar is also helpful.



Exploring Programming Languages

8. What are the most popular programming languages for different types of application development?

- Visual Basic, C/C++, and Java are languages that enable programmers to include control features such as scroll bars, title bars, text boxes, buttons, and expanding and collapsing menus.
- Objective C is a language used in programming applications for mobile devices using iOS and applications that will run under OS X.
- Programmers use HTML tags to structure web pages. HTML5 includes more advanced tags like <video>.
- For more complex web development, scripting programs such as JavaScript, PHP, and VBScript are popular.
- AJAX is a programming solution that uses a combination of technologies to create websites that can update without the user refreshing the page.
- XML and JSON allow programmers to create their own tags so that web pages can exchange information, not just formatting details.

Be sure to check out the companion website for additional materials to help you review and learn, including a Tech Bytes Weekly newsletter—pearsonhighered.com/technaction. And don't forget the Replay Videos .

key terms //

- Active Server Pages (ASP) **435**
AJAX (Asynchronous JavaScript and XML) **436**
algorithm **411**
architecture neutral **434**
assembly language **420**
base class **417**
beta version **426**
binary decision **414**
C **433**
C++ **433**
C# **434**
class **416**
code editing **425**
coding **420**
comment **421**
compilation **423**
compiler **423**
control structure **415**
data **411**
data-flow diagram **409**
data type **419**
debugger **426**
debugging **425**
decision point **414**
derived class **417**
documentation **426**
dynamic decision making **435**
editor **425**
error handling **412**
executable program **423**
eXtensible Markup Language (XML) **437**
fifth-generation language (5GL) **421**
first-generation language (1GL) **420**
flowchart **413**
For **418**
fourth-generation language (4GL) **420**
general availability (GA) **426**
Hypertext Markup Language (HTML) **435**
if else **418**
information **411**
information system **409**
inheritance **417**
initial value **415**
integrated development environment (IDE) **424**
interpreter **424**
Java **434**
Java applet **434**
JavaScript **435**
JavaServer Pages (JSP) **435**
JSON **437**
keyword **418**
logical error **425**
loop **415**
machine language **420**
method **412**
Next **418**
object **417**
Objective C **434**
object-oriented analysis **416**
operator **419**
PHP (Hypertext Preprocessor) **435**
portability **421**
problem statement **411**
Processing **430**
program development life cycle (PDLC) **411**
program specification **409**
programming **411**
programming language **420**
prototype **433**
pseudocode **413**
rapid application development (RAD) **433**
release to manufacturers (RTM) **426**
reusability **417**
runtime error **425**
scope creep **410**
scripting language **435**
second-generation language (2GL) **420**
service pack **426**
source code **423**
statement **418**
Structured Query Language (SQL) **420**
Swift **434**
syntax **418**
syntax error **418**
system development life cycle (SDLC) **409**
test condition **415**
testing plan **412**
third-generation language (3GL) **420**
top-down design **415**
variable **421**
variable declaration **421**
VBScript **435**
Visual Basic (VB) **433**
visual programming languages **431**
web service **433**

chapter quiz // assessment

For a quick review to see what you've learned, answer the following questions. Submit the quiz as requested by your instructor. If you are using MyITLab, the quiz is also available there.

multiple choice

1. What is the first step of the SDLC?
 - a. Problem identification
 - b. Analysis
 - c. Development
 - d. Design
2. What is the last step of the PDLC?
 - a. Describing the problem
 - b. Debugging
 - c. Coding
 - d. Testing and documentation
3. RTM is the point in the software release cycle when
 - a. real people are brought into the testing process.
 - b. the software is released to programmers.
 - c. the software is released to computer manufacturers who install it on the systems they build.
 - d. a software update is released to the public.
4. Java is considered to be *architecture neutral* because
 - a. it can be recompiled to run on many CPUs.
 - b. it only needs to be compiled once and can run on many different CPUs.
 - c. it can run on many different generations of Windows.
 - d. it can run in any browser.
5. PHP is a scripting language used to
 - a. create applications for iOS devices.
 - b. program for Android smartphones.
 - c. control the automatic generation of HTML pages.
 - d. edit text documents.
6. JSON stands for
 - a. Just Short Of Notice.
 - b. JavaScript Only.
 - c. Java Startup Output Naming.
 - d. JavaScript Object Notation.

true/false

- _____ 1. A logical error can be detected and repaired by the compiler.
- _____ 2. A beta version of software is often released before the official commercial release.
- _____ 3. AJAX allows you to create websites that never need to refresh.
- _____ 4. XML and JSON are both common formats for the exchange of information between web services.

critical thinking

1. Key Classes for Shopping

Think of the last time you had a big shopping trip facing you. What would key classes be if you were going to design a software program that helped you manage your shopping excursion? What data and methods would belong in each class?

2. Programming Languages

Is there one programming language that can outperform all others? How would you select the first programming language to learn? Will the number of programming languages you need to know as a professional programmer keep increasing or settle down to just a few choices?

team time //

Working Together for Change

Problem

Your team has been selected to write a program that tells a vending machine how to make change. The program needs to deliver the smallest amount of coins for each transaction.

Task

Divide the class into three teams: Algorithm Design, Coding, and Testing. The responsibilities of each team are outlined as follows.

Process

1. The Algorithm Design team must develop a document that presents the problem as a top-down design sequence of steps, and another document that uses object-oriented analysis to identify the key objects in the problem. Each object needs to be represented as data and behaviors. Inheritance relationships between objects should be noted, as well. You can use flowcharts to document your results.

2. The Coding team needs to decide which programming language would be most appropriate for the project. This program needs to be fast and take up little memory. Use the web to gather information to defend your selection. You may also consider using a product such as Visual Logic (visuallogic.org) to develop the code for a prototype of the system.
3. The Testing team must create a testing plan for the program, including a table listing combinations of inputs and correct outputs.
4. As a group, discuss how each team would communicate its results to the other teams. Once one team has completed its work, are the team members finished? How would the tools of a site such as SourceForge (sourceforge.net) help your development team?

Conclusion

Any programming project requires teams to produce an accurate and efficient solution to a problem. The interaction of team members is vital to successful programming.

ethics project //

Software That Kills

In this exercise, you'll research and then role-play a complicated ethical situation. The role you play might not match your personal beliefs, but your research and use of logic will enable you to represent the view assigned. An arbitrator will watch and comment on both sides of the argument, and together, the team will agree on an ethical solution.

Problem

The Therac-25 was a computerized radiation-therapy machine. Between 1985 and 1987, six people were killed or badly harmed due to a flaw in its software. The dose that one patient had received was estimated to be in the 15,000- to 20,000 rad range. This is roughly 100 times more than the expected dose, and 15 to 20 times more than a fatal dose. The error-prone software remained on the market for months before the problem was recognized, acknowledged, and solved.

Does this situation indicate criminal conduct? Or is it a situation that needs to be resolved in civil court (for monetary damages)? Is this an example of the price society pays for using complex technology so no blame should be assigned?

Research Areas to Consider

- Therac-25 case resolution
- 2010 Toyota recalls for unanticipated acceleration

- Software errors in avionic software systems
- ACM ethical guidelines

Process

1. Divide the class into teams. Research the areas cited above from the perspective of a software developer, the people impacted, or the arbitrator.
2. Team members should write a summary that provides factual support for their character's position regarding the fair and ethical design and use of technology. Then, team members should create an outline to use during the role-playing event.
3. Team members should present their case to the class or submit a PowerPoint presentation for review by the rest of the class, along with the summary and resolution they developed.

Conclusion

As technology becomes ever more prevalent and integrated into our lives, ethical dilemmas will present themselves to an increasing extent. Being able to understand and evaluate both sides of the argument, while responding in a personally or socially ethical manner, will be an important skill.