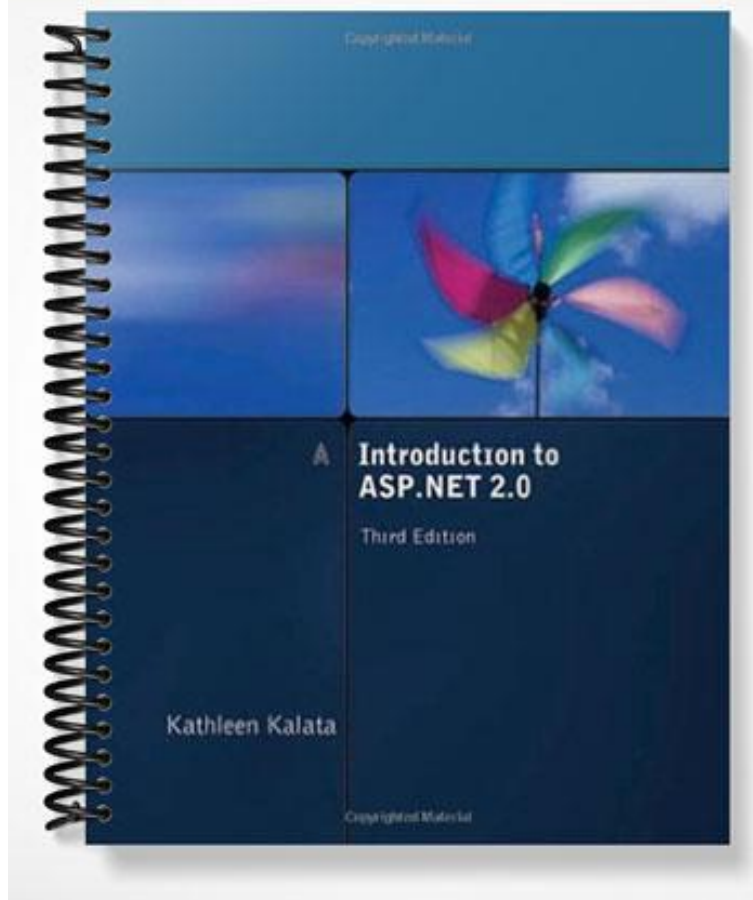


SOLUTIONS MANUAL



Chapter 2

Introduction to Programming

At a Glance

Table of Contents

- Overview
- Objectives
- Instructor Notes
- Quick Quizzes
- Discussion Questions
- Key Terms

Lecture Notes

Overview

ASP.NET code can be written using a variety of programming languages. In this chapter, you will learn how to use object-oriented programming techniques and Visual Basic .NET within an ASP.NET application. You will learn about the .NET Framework, the Visual Basic .NET programming language, and how to create classes, store data using several data structures, access the data using functions and methods using Visual Basic .NET, and debug your application. In this chapter, students will learn the fundamentals for programming a web page using Visual Basic .NET.

Chapter Objectives

In this chapter, students will:

- Integrate programming and Web Forms
- Store data in variables and constants
- Compare data types in Visual Studio .NET
- Work with web objects to store data
- Design .NET web applications
- Create programming structures using Visual Basic .NET
- Use programming control structures
- Locate runtime errors using debugging tools and programming code

Teaching Tips

Your students may have taken another programming logic or language course. They may even have taken a .NET language. You can customize the activities to your class. If they have taken another programming language, then you can simply compare the programming language syntax and methods across each language. Creating a table is very useful.

Instructor Notes

Programming in a new language can be stressful to students. They may have only worked with a programming language for one or two terms. You should realize that they may confuse the syntax and procedural methods across languages. Recommend that students have a single page that contains the common syntax- and language-related commands that are used in the programming language they will work with in this course.

Most beginning programming courses cover this chapter over a period of six to eight weeks. You need to cover the material quickly but thoroughly. Remember that the students will learn the basics well, and then on their projects, they can integrate multiple concepts such as the nesting of decision and loop structures.

Most of the programming required in the book is not extensive. This allows the book to be used with a class where students may be at a variety of levels. You will always have a few students who are further ahead and a few who struggled in their early programming courses.

- Advanced students will want to move ahead quickly. Tell your students that the case studies are where they can combine their knowledge of programming and ASP.NET into more complex applications.
- Advise beginning students to work through each in-chapter exercise and as many of the end-of-the chapter exercises as possible.

Special Discussion Thread Pages

You may want to have a discussion thread for specialty topics for both online and face-to-face courses. For example, you may want to have a discussion thread for content related to:

- Programming in C#
- Programming in Visual Basic .NET

Integrating Programming and Web Forms

Today, many companies expect their programmers to create their web applications. As you learned in the previous chapter, building a web application can be more complicated than building a traditional desktop application. Companies want web programs that can be easily maintained and integrated with other applications. Companies need to build reusable programs with different data sources. To accomplish this, programmers use object-oriented programming techniques to encapsulate the business logic. Then, they create applications that can access the business logic, regardless of how the program was written.

Teaching Tip	<p>Do not assume your students have the same knowledge, skills, and experiences. You may want to give a pre-quiz to determine the students' level.</p> <p>Even if your students are strong programmers, they may not be familiar with the web programming model. Therefore, it is recommended to include this chapter as required reading for all courses.</p>
---------------------	--

Where to Place Your Programming Statements

ASP.NET code can be written using a variety of programming languages. In this chapter, you will learn how to use object-oriented programming techniques and Visual Basic .NET within an ASP.NET application. In this chapter, you will learn about the .NET Framework, the Visual Basic .NET programming language, and how to create classes, store data using several data structures, access the data using functions and methods using Visual Basic .NET, and how to debug your application.

Teaching Tip	<p>This course can be taught using Visual Basic or C#. All exercises in the book can be easily converted to C#.</p>
---------------------	---

The Web Form is a web page that can contain the HTML controls, client-side scripts, and ASP.NET controls. Web Forms contain compiled server programs, which can be written inline with the Web Form or written in a separate file known as the code behind the page. Only server controls can interact with the code behind the page. The filename for the code behind the page has the extension `.aspx.vb`, if the code is written in Visual Basic .NET. Separating the server programming and Web Form allows you to alter your presentation tier without interfering with how the page is processed by the business programming code. Because of this separation of presentation and business logic, the graphic designer could work on the web page layout at the same time the web programmer is modifying the server code.

Teaching Tip	<p>You will want to teach how to insert code in the Web Form inline and in the code behind the page. However, you will want to standardize on one format for your class. This will help make it easier for you to grade and compare student work.</p>
---------------------	---

Teaching Tip	Many books make the mistake of only programming inline in the Web Form page. However, students need to learn how to build reusable code, as they will eventually want to transition to a more object-oriented and scalable programming model. You should encourage students to learn to write the code in the code behind the page and later in classes, which can be reused across an application.
---------------------	---

Configuring the Page Directive

The Page directive allows you to set the default properties, such as the default language for the entire page. The default Language property is set to VB to indicate that the code behind the page is written in Visual Basic .NET. The file location is identified by the CodeFile property. The Inherits property indicates that the code behind the page inherits the partial class.

Teaching Tip	The exercise on where to place your programming statements will help the students learn how to program inline in the web page or in the code behind the page.
---------------------	---

Line Continuation Characters

When the code is very long, use an underscore to continue the code to the next line. It is important not to break the contents of a string across multiple lines, unless each string is contained within its own pair of quotation marks and the string is concatenated.

Teaching Tip	Do not hesitate to teach your students differences in syntax between C# and VB while you are teaching one language. Although it is difficult for many students to learn both languages during one course, learning about the differences will help them know that they can easily move to the other language after they have mastered basic ASP.NET.
---------------------	--

Using Variables and Constants in a Web Form

A variable declaration consists of three parts: the declaration keywords, a variable name, and a data type. The scope identifies what context of the application can access the variable.

Teaching Tip	Depending on the prerequisite course, you may need to spend more time on data types and defining variables and constants, as well as the scope of module-level and local variables.
---------------------	---

If you have multiple variables to declare, you can declare each one on a separate line or use a single line. By default, you must declare all variables before the variables can be used. The Explicit property of the Page directive is used to configure the page to require you to declare all variables before they are used.

Student Activity	Have the class together make a list of words that “should not” be used as variable or constant names.
-------------------------	---

You can use a constant instead of a variable when you want to assign a value that does not change. For example, you can use constants for tax rates, shipping fees, and values used in mathematical equations. The `Const` keyword is used to declare a constant. The naming rules for variables also apply to constants. However, the names of constants are usually all uppercase, as shown in the code that follows this paragraph.

Student Activity	<p>Many instructors include questions on quizzes and the final related to the names of variables and constants.</p> <p>It is useful to have students perform some self-assessment during the course.</p> <p>Ask students to list the words to avoid in the variable, constant, or procedure name. Include words that would be acceptable. Ask students to follow conventions you prefer for capitalization. Remember VB is not case sensitive, but C# is. Include some examples of improper and proper case use.</p> <p>Then, use a software tool such as StudyMate (www.studymate.com) to create Flash-based exercises where students can practice selecting the appropriate or inappropriate names.</p>
-------------------------	--

Working with Different Data Types

This section is typical content that you would find on data types in a programming course. The `String` and `DateTime`, `ArrayList`, and `HashTable` are types and collections you will want to spend more time on in class.

In Visual Basic .NET, strings can be manipulated using different methods that are built into the `String` object. Concatenation is the process of joining one or more strings. The ampersand is used to concatenate the expression to one single string. You can also use the plus sign (+) for concatenation of strings.

As you may recall, the `Format` method of the `String` class allows you to format the numeric value. In general, there are two ways to use the `Format` method. You can use the `Format` method with predefined format styles listed below, or you can create your own user-defined format style.

Teaching Tip	<p>Students need to be encouraged to learn how and when to combine programming statements and concatenating values. You may want to provide several examples of concatenation and conversion in addition to the book samples. There is also a list of methods associated with the string object that is useful for students to refer to when formatting strings and dates with various built-in methods.</p>
---------------------	--

The `DateTime` data type is a .NET data type located in `System.DateTime` and used to store dates and times. The date value is enclosed within a pair of pound signs. You can display the date in a `DateTime` object based on local time, or use the `UtcNow` property to display the Coordinated Universal Time. You can use the `Format` method of the `String` class to format the date and time. To build a custom format for the date and time, you can use masks to represent the values for the day, year, month, minutes, and the time of day.

You can use the `Format` method of the `String` class to format the date and time. You can use a predefined format by passing a standard mask to the `Format` method with the `DateTime` object. For example, the `d` and `D` masks represent the short and long date formats. You can use `t` or `T` masks to represent the short and long time formats. The `r` or `R` masks will display the date as Coordinated Universal Time (UTC).

Although there are several types of collections used in the .NET Framework, you will find the ArrayList and HashTable useful for students who are learning to work with collections and web pages. In more advanced courses, you may want to spend additional time on the other types of collections and the differences and when to use them.

Binding Collections to Web Controls

You can easily bind controls to a collection with your code behind the page. If you want to assign the values in the collection to a TextBox control, you can retrieve the values using a For Next loop, and assign the values to a variable. Then, you would assign the variable to the Text property of the TextBox control.

Check boxes, radio buttons, and drop-down lists allow you to select from lists of options. Remember that with .NET, all of the collections will start with 0 instead of 1. If nothing has been selected, the SelectedIndex property returns the value -1.

Student Activity	<p>In the exercise in the book, you will add items to a Web control in a Web Form using the Add method. You will also add items using an ArrayList and HashTable collection, which will then be used to insert the items into a control in the Web Form.</p> <p>Do not skip these exercises. They layer beginning concepts that will be later used when the student builds web database applications.</p>
-------------------------	---

Quick Quiz

- The _____ tags or the <script runat="server"></script> block tags are used to indicate inline server code within a Web Form.
Answer: <% %>
- The _____ property indicates that the code behind the page inherits the partial class.
Answer: Inherits
- A variable declaration consists of three parts: the declaration keywords, a variable name, and a _____.
Answer: data type
- _____ variables are defined in the class but outside of the procedures, and can be used by any of the procedures within the page.
Answer: Module-level
- _____ defines variables that can be used only by the procedure in which they were declared.
Answer: Protected
- The _____ property of the Page directive is used to configure the page to require you to declare all variables before they are used.
Answer: Explicit
- The _____ keyword is used to declare a constant.
Answer: Const
- The _____ method allows you to join strings together without using the concatenation operator.
Answer: Concat
- _____ is the process of joining one or more strings.
Answer: Concatenation
- The _____ data type is a 32-bit whole number and is the preferred data type for representing a 32-bit number on a 32-bit operating system.
Answer: Integer
- The _____ method of the String class allows you to format the numeric value.
Answer: Format
- The String Format method code to format the number as currency as 1234.45 is _____.
Answer: System.String.Format("{0:C}", 1234.45)
- The String Format method code to format the date and time as 9/9/2007 17:34 is _____.
Answer: Format(MyDT, "m/d/yyyy H:mm")

14. The _____ is a collection object that stores each item in sequential order, and each item is indexed with a number.
Answer: ArrayList
15. The ArrayList is _____, which means that the counting of items in the ArrayList starts at 0 and not 1.
Answer: zero-based

Discussion Questions

Placement of Server Programs

- Compare the benefits of programming server code inline, in block in the Web Form, with the code behind the page.

Visual Basic .NET

- Pick one programming language that you have used before and compare the syntax and data types with Visual Basic .NET.

Binding Data from a Collection to a Server Control on a Web Form

- After completing the example in the book, discuss the problems with binding data that may have occurred. If you did not have any difficulty, list three possible uses for binding data to controls in a web page.

Additional Discussion Questions

- Compare three differences in programming using inline statements or the code behind the page.
- Compare five programming features with Visual Basic .NET to C# or Java.
- The scope identifies what context of the application can access the variable. Explain the difference between local and modular scope. Explain the difference between the keywords used to create variables. (i.e., Dim, Public, Friend, Protected)
- Identify five examples of how constants could be used in a business web application.
- Describe how to use three of the string methods (i.e., Ltrim, Rtrim, Trim, Asc, Chr, Replace, Split, Concat, Compare, LCase, and UCase methods). Provide three reasons to use each method in a business application.
- Compare and contrast three different loop structures. Explain how the iteration occurs in the three loops. Provide an example for each of the three loops (i.e., While loop, Do While loop, Do Until loop, For Next loop, For Each loop).
- Read the documentation on the Systems.Collections namespace. Compare two different types of collections (i.e., ArrayList, HashTable, SortedList, Queue, and Stack).
- Describe three methods that can be used to format a string.
- Describe four of the built-in mathematical methods. Provide concrete examples of how they can be applied in a business web site.

Working with Web Objects to Store Data

HTTP is one of the protocols used to send and receive data over the Internet. With each request from the browser and response from the web server, certain data is transmitted in the header. The header is the portion of a data packet that describes how and where to send the file. The header includes the date and time, the type of request (Get or Post), the page requested, and the HTTP version. Additional data includes the default language, the referring page, the IP address of the client, the cookies associated with that domain, and the user agent. The user agent can be used to identify the client software. Additional data can be sent with the header, such as the values entered into a form. You can programmatically access the information that is transferred with the request and response using various programming objects.

Teaching Tip

Some of these concepts, such as HTTP and QueryString, are not isolated to ASP.NET technology. You should point out that ASP.NET allows you to interact with these built-in objects, which are built into the web server.

Using the Request Object to Retrieve Data from the Header

HTTP collects information and stores the values as a collection of server variables. The QueryString can contain only valid URL characters, has a fixed limit length, and is not appropriate to be used to send sensitive or personal information.

The Request object of the Page class allows you to retrieve any data sent by the browser, including the names and values of form fields and the server variables.

Server variables can be retrieved by using an HttpRequest object with Request.ServerVariables("VARIABLE_NAME"). The server variable name is always in uppercase. The System.Web.HttpRequest class is mapped directly to the Request property of the Page object. So, instead of using the server variables to retrieve the header data, you can use Page.Request.PropertyName or Request.PropertyName. To retrieve the URL requested, IP address of the client, physical path to the file, and domain name of the previous page that contained the link to the current page, use Request.Url, Request.UserHostAddress, Request.PhysicalPath, and Request.UrlReferrer.Host.

Teaching Tip	Do not get caught into teaching new technology in lieu of the old technology. You will want to teach both methods to students because they may need to support older applications in real-world applications.
-------------------------	---

The Response object has several methods to send content to the browser. The WriteFile method allows you to send the entire contents of a text file to the web page. The Write method allows you to send a string to the browser. You can send text, HTML tags, and client script to the browser using the Write method.

The Response object allows you to redirect the browser to another page. You can use the Response object to create server-side redirection. Because this redirection occurs on the server, the visitor never knows that he or she has been redirected to a new page. Although technically the browser is redirected using the HTTP headers, it is not classified as client-side redirection, so the user cannot stop the page redirection. Another method that you can use to change to a new page is Server.Transfer.

Accessing Form Field Data

If you are processing Server controls on the web page, you can access the value properties of the form field directly from your web page code. The property name that you use to access the value is different for each type of Server control. Note, however, that accessing the property value works only with Server controls.

Cross-page posting allows you to send data from one form to another. Essentially, you set thePostBackUrl property of an ImageButton, LinkButton, or Button control to the new page. You can detect if the user came from another page by determining if the Page.Previous property is null.

Session Objects

User information can be tracked across user sessions and across the entire application. Some of this information is retrieved from the properties of the Session object, such as the SessionID, which is a unique identifier that identifies each session. Although the SessionID is determined by several factors, including the current date and the IP addresses of the client and server, it is technically possible—though improbable—for the SessionID to repeat. You cannot change the value of the SessionID property. A special session cookie is used to maintain the session information. When the session ends, the cookie is discarded. If the user opens a new session, a new SessionID is issued. Therefore, a SessionID can be used to track a user across a single session, but not across multiple sessions. To track a user across multiple sessions, the SessionID can be stored in a client-side cookie or in a database that contains the identity of an individual. Some of the information that web sites track is retrieved from forms submitted by users.

The Session object is used to maintain session state across a single user's session. The session variables are also stored in the server's memory. When the session ends, the session variables are released from memory. You can save your state information in session variables or directly within the `__VIEWSTATE` form field.

Teaching Tip	Students have a lot of fun with the Session object. They do not have to have special programs or permissions to create session variables. However, they should know that in real-world applications, it is important that session and application data be securely sent over the Internet. If the session identification is sent with the URL, you may want to make sure to encrypt the session with HTTPS as the protocol to request the web pages.
---------------------	--

Designing .NET Web Applications

Web Application Architecture

Today, even if the entire application is stored on one computer box, the application can still be designed with a tiered framework where the presentation, business, and data layers are separated. The presentation tier contains the user interface. The business programming logic, such as the formulas to calculate interest rate, are implemented in components at a middle tier. It is useful to create the business components to isolate the programming logic so that changes within the visual layer would not impact the programming code. The data tier often consists of one or more data servers such as SQL Server. Scaling may require adding more web servers to handle the requests, to deliver applications, and to store and cache data.

Classroom Activity	<p>Show students an example of a real web-application architecture map. There are many types of software programs that can help diagram a web site map.</p> <p>It is important to help students understand that the physical layout of the files and folders and data is separate from the logical layout, which the end user sees. Later on, when they use the navigation controls, this will be more apparent.</p> <p>Include a discussion question early, asking students to provide a description of what they plan for their final project, for both physical and logical layout and design. This helps to determine if the projects are appropriate, to prevent duplication of projects, to provide feedback to students early, and also helps guide them to starting the projects early.</p>
---------------------------	---

Organization of Classes in the .NET Framework

Your web application will reference classes that you create or that are built into the .NET Framework. The base class libraries are groups of commonly used built-in classes stored in executable files. The base class libraries are organized into hierarchical logical groupings of code called namespaces.

Classroom Activity	Show the student how to locate the documentation for the classes in the MSDN library or web site. The Object Browser can be used to display the classes, properties, and methods built into the namespaces within the .NET Framework.
---------------------------	---

Quick Quiz

1. The _____ is the portion of a data packet that describes how and where to send the file.
Answer: header
2. The _____ object of the Page class allows you to retrieve any data sent by the browser, including the names and values of form fields and the server variables.
Answer: Request
3. The _____ provides a string that can be used to detect the browser version.
Answer: user agent
4. With the _____ property of the Page class, you can directly access the browser application's name and version without the user agent.
Answer: Browser
5. To retrieve the value of a field named CompanyName, you would enter _____.
Answer: Request.Form("CompanyName")
6. If the value of the form method identified in the HTML Form element is Get, then the form results are HTML encoded and appended with a question mark to the URL requested as a single string called the _____.
Answer: QueryString
7. _____ allows you to send data from one form to another.
Answer: Cross-page posting
8. The _____ method allows you to send the entire contents of a text file to the web page.
Answer: WriteFile
9. _____ is used to redirect the visitor to another page at www.course.com.
Answer: Response.Redirect("http://www.course.com/") or Server.Transfer("http://www.course.com/",true)
10. _____ is a unique identifier that identifies each session.
Answer: SessionID
11. The business programming logic, such as the formulas to calculate interest rate, are implemented in components at a _____.
Answer: middle tier
12. Making the application support a larger number of visitors is called _____.
Answer: scaling
13. The _____ are groups of commonly used built-in classes stored in executable files.
Answer: base class libraries
14. The base class libraries are organized into hierarchical logical groupings of code called _____.
Answer: namespaces

Discussion Questions

Processing Web Form Data with Server Objects

- Compare the methods to retrieve values directly from form fields and from the QueryString. Discuss when you cannot access the values of the form fields directly and might need to use the QueryString. Discuss the security risks with using the QueryString.

Sending Content to the Browser with the Response Objects

- Compare the use of Response.Write and Response.WriteFile. Discuss when you might use the WriteFile method as opposed to creating a unique file on the web server.

Server Variables

- List three server variables and identify possible uses for the information within a web application. Compare the use of session data and the ViewState. If the web server crashes, will both data sets be retrievable, and why or why not? Discuss how information can be used to track visitors across web sites.

Additional Discussion Questions

- Explain the format of a QueryString and how to retrieve the form field names and values from a QueryString collection. Visit several web sites. Locate at least five examples of QueryString in real applications online.
- Compare Server.Transfer, Server.Redirect, and Location.href (from your knowledge of JavaScript programming).
- What is the difference between the ViewState and Session data? How can your choice impact the performance of your web application as perceived by the end user?
- Comparing the logical tier and physical architecture in enterprise environments. Explain how companies build scalable web applications using n-tiered hierarchical models.

Programming with Visual Basic .NET

Decision control structures used conditional expressions to determine which block of code to execute. The block of code is a group of one or more programming statements. If a condition is met, then a block of code is executed; otherwise, an alternate block of code is executed. This type of programming was often referred to as procedural programming.

Teaching Tip	Again, the depth that you will cover this in the class depends upon the background of your students. You may want to give them an exercise to help determine their programming knowledge and skill level.
---------------------	---

Creating a Class in ASP.NET

You create custom objects based upon the object definition, and then you can access the procedures and properties of the object across multiple web pages. An object is a set of related, compartmentalized code that is based upon a class. In addition to the Page class, you can create a user-defined class with no visual component but with global variables, properties, and procedures. The first task is to create the object definition, also called the base class or class definition. The class is not accessed directly; it is the code that will be used to create the procedures and properties of the object. Once you define the class, you can create many objects based upon the class definition. In other words, the class is the template for the new object. When you create an object, you are really creating an instance of the class. Instantiation is the process of declaring and initializing an object from a class. The source file for the class ends in .vb, and by default, is placed in a directory named App_Code. You can create more than one class in the source file. You can restrict which applications can have access to the class using these additional keywords: public, private, protected, and friend.

Using a Procedure

Some programming statements are action statements; for example, they change the value of a text box. Some programming statements are decision control structures, which allow you to organize the order in which the code is executed. Procedures are used to contain one or more programming statements. Procedures are executed only when they are called by another procedure or when an event occurs. Procedures always execute in the order in which they are called, not the order in which they are created. Parameters can be passed in a comma delimited list within a pair of parentheses. You must also pass the type of data stored in the parameter. If no arguments are passed, you use an empty pair of parentheses.

Creating Subprocedures

Subprocedures are declared using the keyword Sub. You can use the Exit Sub statement to stop the subprocedure. The subprocedures are executed in the order they are called by specifying the name of the subprocedure or using the keyword Call and passing any parameters required. You can pass the values as variables containing values, or use the value.

Creating Event Procedures

An event procedure is a type of subprocedure that is attached to an object with events. An event procedure is not executed until triggered by an event. The event procedure is the code that will execute when the event handler intercepts an event, such as a button click. Note that the event procedure is not called, but rather is triggered, when the event occurs for that specific object. The object calls the event procedure when it detects that the event has occurred.

Event procedure names are based upon the name of the object and the event name. An underscore (_) is used to separate the object name and the event name. The Sender As Object parameter identifies the object in the page that initiated the event. The EventArgs, represented as an e, is the set of parameters that was sent with the object. An event procedure does not return a value.

Page Events

The page lifecycle is the order that the Web Form is processed on the server and rendered in the browser. Page-related events will occur as part of page lifecycle. For example, the Page_Load event is triggered when the page is loaded into memory on the server. You can intercept the event handlers for the page events within code behind the page by writing an event procedure. The Page_Init event initializes the page framework hierarchy, creates all of the controls, deserializes the ViewState, and applies the previous settings to the controls. The Page_Load event loads any Server controls into memory and occurs every time the page is requested. You can determine if the page has previously been loaded by using the Page.IsPostBack property. You can view the order in which the Server controls are rendered to the web page by turning on the Trace feature. The Trace feature allows you to view the page controls, the order in which they are loaded, and the time it takes to load each control. To turn on the Trace feature, set the Trace property in the Page directive to true.

Creating Functions

There are hundreds of built-in functions available within the .NET Framework including mathematical, date and time, string, and formatting functions, among others. You can also create your own function. You can leave the function at any time by inserting the Exit Function statement. Functions return a value—one value only—to the program that called the function by using the Return statement. The data type returned is identified in the function declaration.

Creating a Property Method

A Property method sets the value of a variable defined within an object and is often used to expose private variables that are defined within the class. Some properties are assigned to the object by default within the object definition. Then, all new objects inherit the same properties as the original object definition. However, if an object does not assign a value to a property, and a new object is created without assigning a value to the property, the property is assigned the default value “undefined.” You can also assign values to properties that are declared in the class when you create the object from the class and after the object is created.

Teaching Tip	Again, the depth that you will cover this in the class depends upon the background of your students. You may want to give them an exercise to help determine their programming knowledge and skill level.
-------------------------	---

How Web Applications Interact with the Systems.Drawing.Graphics Class

You can do many fun exercises with students that will help them understand how to interact with the wide variety of classes and procedures available to them.

The Systems.Drawing.Graphics class is referred to as the GDI+ or simply the Graphics class. With the Graphics class, you can create visual objects such as lines and shapes like rectangles and ellipses; paint the background; include and scale existing images; build custom objects; and even insert charts into your web applications.

Image formats most widely supported on the web include JPEG, GIF, and PNG. The Graphics class now has the ability to save the image to the HTML output stream as other formats, such as JPEG, by using the Save method.

Dynamically Creating Server Controls

The Placeholder control is used as a container to store dynamically added Server controls. The Placeholder control does not produce any visible output without the use of other controls. To add, insert, or remove a control from the Placeholder control, you can use the Control class of the control namespace. To do this, you must know how to use the constructor to create a control dynamically. For example, say you create and initialize a new instance of the Button class. This won't display anything. You have to add it to the Controls Property collection of the Placeholder class. Then, the page will display the control in its place, and format the Button properties.

The Panel control can contain other controls and creates a DIV tag to enclose the contents. You can set properties such as wrapping, absolute positioning, font type, and scroll bars. For example, if you want to add some text to a panel, simply add a Label control within the panel tags or enter the text directly into the panel. The Label control creates contents within a Panel control using the SPAN tag. You use the Text property to display text in a Label control.

You can also use a Literal control to write content directly to the page. Literal controls are often used to add client-side HTML and text to the page. Any non-server code, such as HTML tags or tags without an ID property assigned, is placed within Literal controls when the page is rendered.

The HyperLink control is used to create an anchor tag that can link to another page or a target by using the ImageURL property. The displayed text is configured using the Text property. The Text property also becomes the ToolTip by default. The ToolTip is a new property that is used to display a message when the user places the pointer over the object. You can change the Target property of the hyperlink. The target is the window or frame used to load the web page linked to when the HyperLink control is clicked. The default value for the target is String.Empty. You can change the target to any named window or one of the reserved window names. The named window must begin with a letter in the range of a through z and is case sensitive.

Programming Control Structure

Control statements allow you to use conditions to determine both whether and how an action statement is executed, and the order in which action statements are executed. You can nest statements within other programming control structures.

Decision control structures allow you to alter the execution order of action statements on the basis of conditional expressions. A conditional expression is an expression that is evaluated by the program as true or false. The If Then and Select Case statements are examples of decision control structures.

Loop structures allow you to repeat action statements on the basis of conditional expressions. While, Do While, For Next, and For Each are examples of loop structures. You can also nest any of these programming control structures. Using a loop structure, you can repeat action statements or control statements any number of times. The Loop structure repeats a block of statements while a Boolean condition is True or until the condition becomes True. Each repetition of a loop is called iteration. If you never update the loop index variable, the loop will continue to run without a natural ending; this is called an infinite loop. The loop uses a conditional expression to determine when to stop. The loop stops when the conditional expression is true.

Teaching Tip	Again, the depth that you will cover this in the class depends upon the background of your students. You may want to give them an exercise to help determine their programming knowledge and skill level.
---------------------	---

Introduction to Web Configuration and Debugging

Debugging is the process of detecting programming errors within the application. Error handling is a collection of techniques that allow you to identify programming errors in your code, which are often called bugs. Many programming errors are caused by incorrect syntax or spelling errors and can be detected during development using tools such as IntelliSense. However, some errors, such as programming logic errors, are more difficult to detect because they are not detected until runtime, when the application is running. To handle errors that arise during program execution, you can write code that interprets error messages and executes when an error is detected, even generating custom error messages and global error handlers.

Exception Handling

An exception is an object that is thrown when a predefined runtime error occurs. The Try-Catch-Finally statement is the structure that is used to handle these exceptions. The .NET System contains an Exception class that acts as the base class for all exceptions. These predefined exception errors are categorized in the SystemException class and ApplicationException class. You can also use a general exception handler to catch general exceptions.

Common Error Status Messages

This status message indicates a successful or unsuccessful response from the web server. If the resource is received and returned, then a Status Code 200 is returned to the browser in the header. Table 2-2 lists common status codes exposed by the HttpStatusCode property of System.Net. These are the same status codes that are used by the web server that is running the HTTP protocol. You can retrieve this information using the HttpStatusCode property.

Creating a Custom Error Message

The ErrorPage property of the Page directive can be used to identify an error page that is displayed when an error occurs on that page. When an error occurs, the client is redirected to this error page. You can customize the error messages for the entire web site by using the customErrors node in the web.config file. You can edit this file manually or use the Web Site Administration Tool (WSAT).

Configuring the Web Application to Use the Debugger Tool

One of the most difficult problems programmers traditionally faced was debugging their web applications. A debugger tool allows you to step through the application at breakpoints, which are defined points in the program where processing stops so that you can view the variables, properties, and other application settings. To debug your application, you must first enable the page to support debugging. The Page directive includes an attribute named Debug, which allows you to turn on support for debugging. You can also turn on debugging for the entire application by configuring debugging in the web configuration file.

Teaching Tip	You should also include a listing of the best practices that you will require your students to comply with when they submit their homework assignments and project.
---------------------	---

Quick Quiz

1. _____ structures used conditional expressions to determine which block of code to execute.
Answer: Decision control
2. A(n) _____ is a set of related, compartmentalized code that is based upon a class.
Answer: object
3. _____ is the process of declaring and initializing an object from a class.
Answer: Instantiation
4. The source file for the class ends in .vb, and by default, is placed in a directory named _____.
Answer: App_Code
5. You can pass zero or more arguments, called _____, to a procedure or other type of programming structure.
Answer: parameters
6. You can use the _____ statement to stop the subprocedure.
Answer: Exit Sub
7. The subprocedures are executed in the order they are called by specifying the name of the subprocedure or using the keyword _____ and passing any parameters required.
Answer: Call
8. A(n) _____ procedure is a type of subprocedure that is attached to an object with events.
Answer: event
9. The _____ is the order that the Web Form is processed on the server and rendered in the browser.
Answer: page lifecycle
10. The _____ event occurs immediately before the control hierarchy is rendered and sent to the browser.
Answer: Page_PreRender
11. The _____ feature allows you to view the page controls, the order in which they are loaded, and the time it takes to load each control
Answer: Trace
12. You can leave the function at any time by inserting the _____ statement.
Answer: Exit Function
13. A(n) _____ method sets the value of a variable defined within an object and is often used to expose private variables that are defined within the class.
Answer: Property
14. With the _____ class, you can create visual objects such as lines and shapes like rectangles and ellipses; paint the background; include and scale existing images; build custom objects; and even insert charts into your web applications.
Answer: Graphics
15. The _____ control is used as a container to store dynamically added Server controls.
Answer: Placeholder
16. _____ controls are often used to add client-side HTML and text to the page.
Answer: Literal
17. The HyperLink control is used to create an anchor tag that can link to another page or a target by using the _____ property.
Answer: ImageURL
18. A(n) _____ expression is an expression that is evaluated by the program as true or false.
Answer: conditional
19. The _____ class is the base class for all predefined exceptions.
Answer: SystemException
20. If the resource is received and returned, then a Status Code _____ is returned to the browser in the header.
Answer: 200

Discussion Questions

Programming Organization and Controls

- Compare the use of programming structures such as decision control and looping structures. Provide examples of at least two.

Functions, Procedures and Methods

- Programming requires making choices about how to write the application. Compare the use of functions, procedures, and methods. Include when to use them and one example of each. Discuss what determines when user-defined and built-in functions, methods, event procedures and user-defined subprocedures are executed.

Troubleshooting

- Compare the debugging and exception handling tools used to create web applications and Windows applications. Discuss when it would be appropriate to use exception handling in a web application. List at least five things that are displayed when the Trace feature is turned on that can be used to help identify problems in the page or web site.

Additional Discussion Questions

You may want to combine the discussion questions with additional short activities such as Internet Field Trips. The choice that you want your students to submit their answers could vary with each activity. Many of these activities can be submitted with a discussion question or as a separate content item such as a Word document, HTML page, or even as an ASP.NET Web Form.

1. Read about partial classes at [http://msdn2.microsoft.com/en-us/library/bb129230\(vs.80\).aspx](http://msdn2.microsoft.com/en-us/library/bb129230(vs.80).aspx). Describe the benefits of using a partial class.
2. Read about the StringBuilder class. Visit [http://msdn2.microsoft.com/en-us/library/system.text.stringbuilder\(vs.80\).aspx](http://msdn2.microsoft.com/en-us/library/system.text.stringbuilder(vs.80).aspx). Compare the StringBuilder class and concatenating strings manually.
3. Read the documentation in the MSDN Online library on .NET data types and language equivalence at [http://msdn2.microsoft.com/en-us/library/czz35az4\(vs.80\).aspx](http://msdn2.microsoft.com/en-us/library/czz35az4(vs.80).aspx) and [http://msdn2.microsoft.com/en-us/library/k1s94fta\(vs.80\).aspx](http://msdn2.microsoft.com/en-us/library/k1s94fta(vs.80).aspx) and the Common Type System at [http://msdn2.microsoft.com/en-us/library/zcx1eb1e\(vs.80\).aspx](http://msdn2.microsoft.com/en-us/library/zcx1eb1e(vs.80).aspx). Compare at least three data types between Visual Basic .NET and C# (Char, String, Numeric, DateTime, Boolean).
4. View the .NET Framework documentation on the QuickStart web sites and MSDN Online Library. Draw a visual representation of the .NET hierarchical namespace structures that are often used with web programming. Hint: locate the namespaces for the various web objects mentioned in this chapter (i.e., Page, HTML Server controls, Web Controls, Session object, etc.).
5. Draw a diagram with the steps required from requesting a web page until the page is rendered in the browser on the client's computer. Label each step with a number and provide a list describing each step.
6. Programming Best Practices is an important lesson. Visit some of the Microsoft web sites that discuss best practices for web programmers. Visit <http://msdn2.microsoft.com/en-us/library/ms184411.aspx> to learn about exception handling, <http://msdn2.microsoft.com/en-us/library/sa4se9bc.aspx> to learn about security best practices, and <http://msdn.microsoft.com/msdnmag/issues/05/09/SessionState/default.aspx> to learn about best practices to build secure and scalable solutions. Provide a short list of at least five things you can do to implement best practices in web applications.
7. Visit the Language Support on the QuickStart web sites for ASP.NET 1.1 (www.dotnetjunkies.com/quickstart/aspplus/doc/quickstart.aspx). The samples are viewable in C# and Visual Basic .NET. Visit the 2.0 web site (<http://quickstarts.asp.net/QuickStartv20/aspnet/default.aspx>). Identify at least three programming techniques and describe the difference between C# and Visual Basic .Net.

Teaching Tip	<p>Best practices are commonly accepted activities within a given discipline. Remember that web programmers must integrate knowledge from a variety of disciplines such as networking, programming, internetworking, graphic design, database management, computer security, and web development</p> <p>On the next page, there is a table that students can use to compare different syntax across programming languages. It is useful to use the QuickStart web site from ASP.NET 1.1 and 2.0 to compare the languages.</p>
---------------------	---

Programming Language Syntax Chart

Programming Features & Syntax	Language 1	Language 2
Line continuation or ending characters		
Concatenation operator		
Declaring and naming a variable		
Declaring constants		
Char data type		
String data type		
Using the String class format		
Numeric data types		
Built-in mathematical methods		
DateTime data type		
Formatting the date		
Retrieving the current date		
A Boolean data type		
Converting data types		
The ArrayList collection		
The Hashtable collection		
Creating a Class		
Using a procedure		
Creating subprocedures		
Exiting Sub to stop a subprocedure		
Keyword Call		
Passing parameters required		
Creating event procedures		
An event procedure		
Creating functions		
Built-in functions		
Creating a custom function		
Exiting function statement		
Returning a value		
Creating a Property Method		
Systems.Drawing.Graphics Class		
If Then		
Comparison operator		
A logical operator		
Select Case		
While loop		
Do While loops		
Do Until loops		
For Next loop		
For Each loop		
Exception Handling Try-Catch-Finally		
Debugger Tool		
Programming Best Practices		

Key Terms

- **Abstraction:** The ability to create a concept using code
- **ArrayList Class:** Stores each item in sequential order, and each item is indexed with a number
- **Assignment operator:** The equal sign (=)
- **Base Class Libraries:** Commonly used code that provides general functions that can be accessed from any application
- **Boolean:** Type only has two possible values, True value or False value
- **Byte:** Stores an integer between 0 and 255. Short - is a 16-bit number. Therefore, a short data type can only store values from -32,768 to 32,767.
- **Instantiation:** The process of declaring and initializing an object from a class
- **Call keyword:** Can be used to call a function or subprocedure and is optional
- **Char:** Allows you to store a single text value as a number between 0 and 65, 535
- **Class definition:** The code used to create the class
- **Code Behind the Page:** A file that contains the server code that corresponds to the Web Form
- **Collection:** An object that can store other elements
- **Concatenation:** The process of joining one or more strings
- **Constant:** A variable that does not change such as tax rates, shipping fees, and values used in mathematical equations
- **DateTime:** Data type used to store any dates and times between 01/01/0001 and 12/31/9999 as mm/dd/yyyy and hh:mm:ss
- **Data type:** Identifies what kind of data the variable can store
- **Decimal:** Can store numbers up to 28 decimal places and is often used to store currency data
- **Decision control structures:** Allow you to organize the order in which the code is executed
- **Declaring a variable:** The process of reserving the memory space for the variable before it is used in the program
- **Double:** Supports larger numbers than the single data type
- **Encapsulation:** Means that the inner workings of the object, such as the procedures and properties, are maintained within the object so objects or programs are self-contained
- **Event handlers:** Execute code when the event occurs
- **Functions:** A named grouping of one or more programming statements that can return a value to the program that called the function by using the **return** keyword
- **HashTable Class:** Collection creates the index of elements using an alphanumeric key
- **Inheritance:** Allows you to derive the interface and behaviors from another class
- **Integer:** Data type is a 32-bit whole number
- **Long:** A 64-bit number
- **MSIL:** The intermediate language created by compiling the .NET code
- **Namespace:** A hierarchical way to identify resources in .NET
- **.NET Framework:** Additional tools that are installed on top of the operating system that enable the programmer to create applications
- **Option Explicit:** Requires the page to declare all variables before they can be used
- **Option Strict:** Prevents conversion that would result in data loss
- **Parameters:** Values passed to the subprocedures and functions; multiple values are separated in a comma delimited list within a pair of parentheses
- **Panel control:** Contains other controls and creates a DIV tag to enclose the contents
- **Placeholder control:** Inherits from the Control class and is used as a container to store dynamically added server controls
- **Procedures:** Named grouping of one or more programming statements that is executed when the procedure has been called by another procedure or event handler
- **Property:** A method that is used to get and set values
- **Polymorphism:** The ability to create code that can be applied to objects that are derived from different classes
- **Queue Class:** Provides sequential access to the elements but stores them in First In, First Out (FIFO)
- **QueryString Collection:** Contains all the name and value pairs encoded and appended to the URL with a ?

- **Reference types:** Strings, classes, arrays, collections, and objects
- **Return keyword:** Used to identify the value that is returned to the function call.
- **Single:** Represents a single-precision floating point number
- **Script:** A series of commands usually linked or embedded into a web page that when interpreted by a scripting engine, allows web developers to create dynamic web pages
- **String:** Used to store text, which can consist of numbers, characters, and date information.
- **SortedList Class:** Collection also uses a key, but the collection is indexed by both the key and the item
- **Stack Class:** Provides sequential access but stores them in Last In, First Out (LIFO) order
- **Trace Property:** Can be turned on by specifying Trace="true" in the @Page command
- **Value types:** Also referred to as primitive types, structures, or subtypes; include Boolean, Integer, Decimal, Char, and DateTime
- **Web.config:** Application configuration file
- **WinForms:** Visual Basic.NET application forms that interact with the user are inherited from the System.Windows.Forms namespace