Preface

Welcome to Starting Out with Visual C# 2010, Second Edition. This book is intended for an introductory programming course and is ideal for students with no prior experience. Students who are new to programming will appreciate the clear, down-to-earth explanations and the detailed walk-throughs that are provided by the hands-on tutorials. More experienced students will appreciate the depth of detail as they learn about the .NET Framework, databases, and other topics.

As with all the books in the Starting Out With series, the hallmark of this text is its clear, friendly, and easy-to-understand writing. In addition, it is rich in example programs that are concise and practical. The programs in this book include short examples that highlight specific programming topics, as well as more involved examples that focus on problem solving. Each chapter provides numerous hands-on tutorials that guide the student through each step of the development of an application. In addition to detailed, step-by-step instructions, the tutorials also provide the application's completed code and screen captures of the completed forms.

New to This Edition

If you have used the First Edition of Starting Out with Visual C#, you will recognize that this edition has been completely rewritten. Although the friendly presentation and clear writing style remain the same, many changes and improvements have been made, as summarized here:

GUI-Based Approach. This edition uses a visual, GUI-based approach. From the beginning, students learn to develop Windows Forms applications.


New Organization. The book has been completely reorganized to accommodate its new pedagogy of using objects early but writing classes late, and developing GUI applications.

Hands-On Tutorials. Each chapter includes several hands-on tutorials that lead the student through the development of a project.

Lists. Lists are introduced along with arrays.

Exceptions. Simple exception handling is introduced in Chapter 3.

Databases. A new chapter on databases has been added.

A GUI-Based Approach

Beginning students are more motivated to learn programming when their applications have some sort of graphical element, such as a graphical user interface. Students using this book will learn to create GUI-based, event-driven, Visual C# applications. The Visual Studio (or Visual C# Express) environment is used to create forms that are rich with user interface controls and graphical images.

Learn to Use Objects Early, Learn to Write Classes Later

This book explains what objects are very early and shows the student how to create objects from classes that are provided by the .NET Framework. It then introduces the student to the fundamentals of input and output, control structures, methods, arrays and lists, and file I/O. Then the student learns to write his or her own classes and explores the topics of inheritance and polymorphism.
Visual Studio and Visual C# Express Edition

The book can be used with either Visual Studio 2010 or Visual C# 2010 Express Edition. The book is bundled with Microsoft's Visual C# 2010 Express Edition—a streamlined product that captures the best elements of Visual Studio in an ideal format for learning programming. The Express Edition offers an impressive set of tools for developing and debugging Visual C# applications, including those that work with databases and use SQL.

Brief Overview of Each Chapter

**Chapter 1: Introduction to Computers and Programming.** This chapter begins by giving a very concrete and easy-to-understand explanation of how computers work, how data is stored and manipulated, and why we write programs in high-level languages. In this chapter, the student learns what an object is and sees several examples by studying the objects that make up a program's graphical user interface. The chapter discusses steps in the programming development cycle. It also gives an introduction to the Visual Studio or Visual C# Express environment.

**Chapter 2: Introduction to Visual C#.** In this chapter the student learns to create forms with labels, buttons, and picture boxes and learns to modify control properties. The student is introduced to C# code, and learns the organizational structure of namespaces, classes, and methods. The student learns to write simple event-driven applications that respond to button clicks or provide interaction through clickable images. The importance of commenting code is also discussed.

**Chapter 3: Processing Data.** This chapter introduces variables and data types. It discusses the use of local variables and variables declared as fields within a form class. The student learns to create applications that read input from TextBox controls, perform mathematical operations, and produce formatted output. The student learns about the exceptions that can occur when the user enters invalid data into a TextBox and learns to write simple exception-handling code to deal with those problems. Named constants are introduced as a way of representing unchanging values and creating self-documenting, maintainable code. The student also learns more intricacies of creating graphical user interfaces.

**Chapter 4: Making Decisions.** In this chapter the student learns about relational operators and Boolean expressions and is shown how to control the flow of a program with decision structures. The if, if-else, and if-else-if statements are covered. Nested decision structures, logical operators, and the switch statement are also discussed. The student learns to use the TryParse family of methods to validate input and prevent exceptions. Radio buttons, check boxes, and list boxes are introduced as ways to let the user select items in a GUI.

**Chapter 5: Loops, Files, and Random Numbers.** This chapter shows the student how to use loops to create repetition structures. The while loop, the for loop, and the do-while loop are presented. Counters, accumulators, and running totals are also discussed. This chapter also introduces sequential file input and output and using text files. The student learns various programming techniques for writing data to text files and reading the contents of test files. The chapter concludes with a discussion of pseudorandom numbers, their applications, and how to generate them.

**Chapter 6: Modularizing Your Code with Methods.** In this chapter the student first learns how to write and call void methods. The chapter shows the benefits of using methods to modularize programs and discusses the top-down design approach. Then, the student learns to pass arguments to methods. Passing by value, by reference, and output parameters are discussed. Finally, the student learns to write value-returning methods.

**Chapter 7: Arrays and Lists.** Arrays and lists are reference-type objects in C#, so this chapter begins by discussing the difference between value type and reference type objects in the C# language. Then, the student learns to create and work with single-dimensional
and two-dimensional arrays. The student learns to pass arrays as arguments to methods, transfer data between arrays and files, work with partially filled arrays, and create jagged arrays. Many examples of array processing are provided including examples of finding the sum, average, highest and lowest values in an array. Finally, the student learns to create List objects and store data in them.

**Chapter 8: More about Processing Data.** This chapter presents several diverse topics. Now that the student has studied the fundamentals of Visual C# programming, he or she can use the topics presented in this chapter to perform more advanced operations. First, various string and character processing techniques are introduced. Then the student learns to use structures to encapsulate several variables into a single item. The student next learns to create and use enumerated types. Last, the student learns about the ImageList control, a data structure for storing and retrieving images.

**Chapter 9: Classes and Multiform Projects.** Up to this point, the student has extensively used objects that are instances of .NET Framework classes. In this chapter the student learns to write classes to create his or her own objects. The student learns to create fields, methods, and constructors and learns to implement properties. Creating arrays of objects and storing objects in a List are also discussed. A primer on finding the classes in a problem as well as their responsibilities is provided. Finally, the chapter shows the student how to create multiple form classes in a project, instantiate those classes, and display them.

**Chapter 10: Inheritance and Polymorphism.** The study of classes continues in this chapter with the subjects of inheritance and polymorphism. The topics covered include base classes, derived classes, how constructors functions work in inheritance, method overriding, and polymorphism. Abstract classes and abstract methods are also discussed.

**Chapter 11: Databases.** This chapter introduces the student to basic database concepts. The student first learns about tables, rows, and columns and how to create a SQL Server database in Visual Studio. The student then learns how to connect a database to a Visual C# application and display a table in a DataGridView control, a Details view and other data-bound controls. Finally, the student learns how to write SQL Select statements to retrieve data from a table.

**Appendix A: C# Primitive Data Types.** This appendix gives an overview of the primitive data types available in C#.

**Appendix B: Additional User Interface Controls.** This appendix shows how to create a variety of controls such as ToolTips, combo boxes, scroll bars, TabControls, WebBrowser controls, ErrorProvider components, and menu systems.

**Appendix C: ASCII/Unicode Characters.** This appendix lists the ASCII (American Standard Code for Information Interchange) character set, which is also the Latin Subset of Unicode.

**Appendix D: Answers to Checkpoint Questions.** This appendix provides the answers to the Checkpoint questions that appear throughout each chapter in the book.

**Organization of the Text**

The text teaches Visual C# step by step. Each chapter covers a major set of programming topics, introduces controls and GUI elements, and builds knowledge as the student progresses through the book. Although the chapters can be easily taught in their existing sequence, there is some flexibility. Figure P-1 shows the chapter dependencies. As shown in the figure, Chapters 1–7 present the fundamentals of Visual C# programming and should be covered in sequence. Then, you can move directly to Chapter 8, Chapter 9, or Chapter 11. Chapter 10 should be covered after Chapter 9.
Figure P-1  Chapter dependencies

Features of the Text

Concept Statements. Each major section of the text starts with a concept statement. This statement concisely summarizes the main point of the section.

Tutorials. Each chapter has several hands-on tutorials that guide the student through the development of an application. Each tutorial provides detailed, step-by-step instructions, as well as the application’s completed code and screen captures of the completed forms.

Example Programs. Each chapter has an abundant number of code examples designed to highlight the current topic.

Notes. Notes appear at several places throughout the text. They are short explanations of interesting or often misunderstood points relevant to the topic at hand.

Tips. Tips advise the student on the best techniques for approaching different programming or animation problems.

Warnings. Warnings caution students about programming techniques or practices that can lead to malfunctioning programs or lost data.

Checkpoints. Checkpoints are questions placed at intervals throughout each chapter. They are designed to query the student’s knowledge quickly after learning a new topic. The answers to the Checkpoint questions can be found in Appendix D.

Review Questions. Each chapter presents a thorough and diverse set of Review Questions and Exercises. They include Multiple Choice, True/False, Algorithm Workbench, and Short Answer.

Programming Problems. Each chapter offers a pool of Programming Problems designed to solidify the student’s knowledge of the topics currently being studied.

Supplements

Student. The following supplementary material is bundled with the book:

- A student CD-ROM containing the source code and files required for the chapter tutorials.
- A DVD containing Microsoft Visual C# 2010 Express Edition
Instructor. The following supplements are available to qualified instructors:

- Answers to all Review Questions in the text
- Solutions for all Programming Problems in the text
- Completed versions of all tutorials
- PowerPoint presentation slides for every chapter
- Test bank

For information on how to access these supplements, visit the Pearson Education Instructor Resource Center at www.pearsonhighered.com/irc or e-mail computing@pearson.com.

Acknowledgments

I would like to thank my family for their love and support in all my many projects. Thanks also go to Chris Rich for his work on the supplements. I am extremely fortunate to have Michael Hirsch as my editor and Stephanie Sellinger as editorial assistant. Michael’s support and encouragement make it a pleasure to write chapters and meet deadlines. I am also fortunate to have Yez Alayan as marketing manager, and Kathryn Ferranti as marketing coordinator. They do a great job getting my books out to the academic community. I had a great production team, led by Jeff Holcomb, Managing Editor, and Heather McNally, Production Project Manager. Thanks to you all!

About the Author

Tony Gaddis is the principal author of the Starting Out With series of textbooks. Tony has nearly 20 years experience teaching computer science courses at Haywood Community College in North Carolina. He is a highly acclaimed instructor who was previously selected as the North Carolina Community College Teacher of the Year and has received the Teaching Excellence award from the National Institute for Staff and Organizational Development.

The Starting Out With series includes introductory books using the C++ programming language, the Java™ programming language, Microsoft® Visual Basic®, Microsoft® C#®, Python, Programming Logic and Design, and Alice, all published by the Addison-Wesley imprint of Pearson Education.