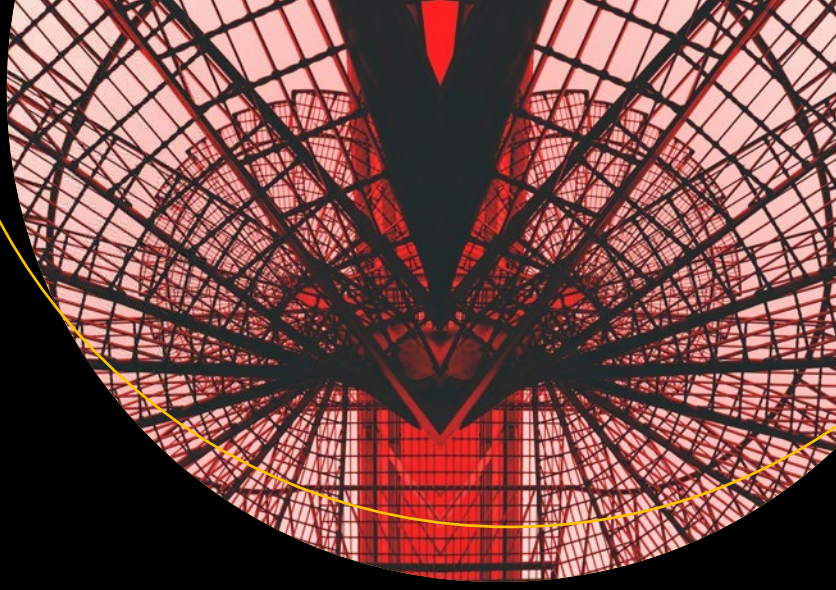

Design Patterns in C#

A Hands-on Guide with Real-world Examples

—

Second Edition

—

Vaskaran Sarcar

Foreword by Priya Shimanthoor

Apress®

Design Patterns in C#

A Hands-on Guide with Real-world Examples

Second Edition

Vaskaran Sarcar
Foreword by Priya Shimanthoor

Apress®

Design Patterns in C#: A Hands-on Guide with Real-world Examples

Vaskaran Sarcar
Garia, Kolkata, West Bengal, India

ISBN-13 (pbk): 978-1-4842-6061-6 ISBN-13 (electronic): 978-1-4842-6062-3
https://doi.org/10.1007/978-1-4842-6062-3

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Smriti Srivastava
Development Editor: Laura Berendson
Coordinating Editor: Shrikant Vishwakarma

Cover designed by eStudioCalamar

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-6061-6. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

This book is dedicated to all the unsung heroes and volunteers who are continuously fighting at the frontlines of the COVID-19 battle to save humanity and this beautiful world.

Table of Contents

About the Author

Vaskaran Sarcar obtained his master's degree in software engineering from Jadavpur University, Kolkata (India), and an MCA from Vidyasagar University, Midnapore (India). He was a National Gate Scholar (2007–2009), and he has more than 12 years of experience in education and the IT industry. Vaskaran devoted his early years (2005–2007) in teaching at various engineering colleges. Later, he joined HP India PPS R&D Hub Bangalore, where he worked until August 2019. At the time of his retirement from the IT industry, he was a Senior Software Engineer and Team Lead at HP. Following his passion, Vaskaran is now an independent full-time author. His books include the following:

- *Getting Started with Advanced C#* (Apress, 2020)
- *Interactive Object-Oriented Programming in Java Second Edition* (Apress, 2019)
- *Java Design Patterns Second Edition* (Apress, 2019)
- *Design Patterns in C#* (Apress, 2018)
- *Interactive C#* (Apress, 2017)
- *Interactive Object-Oriented Programming in Java* (Apress, 2016)
- *Java Design Patterns* (Apress, 2016)
- *C# Basics: Test Your Skills* (Createspace, 2015)
- *Operating System: Computer Science Interview Series* (Createspace, 2014)

About the Technical Reviewers

Carsten Thomsen is a back-end developer primarily but works with smaller front-end bits as well. He has authored and reviewed several books and has created numerous Microsoft Learning courses on software development. Carsten works as a freelancer/contractor in various countries in Europe; Azure, Visual Studio, Azure DevOps, and GitHub are some of his favorite tools. An exceptional troubleshooter, he asks the right questions, including the less logical ones, in the most logical to least logical fashion. He also enjoys working with architecture, research, analysis, development, testing, and bug fixing. Carsten is a communicator with skills in mentoring, team leadership, research, and presenting new material.

Shekhar Kumar Maravi is a lead engineer in design and development whose main interests are programming languages, algorithms, and data structures. He obtained his master's degree in computer science and engineering from the Indian Institute of Technology, Bombay. After graduation, he joined Hewlett-Packard's R&D Hub in India to work on printer firmware. Currently, he is a technical lead engineer at Siemens Healthcare's R&D division. He can be reached by email at `shekhar.maravi@gmail.com` or via LinkedIn at `www.linkedin.com/in/shekharmaravi`.

Foreword

Written programs need to be flexible, easily maintainable, and reusable. How do we know that a program is as elegant as it can be? The answer is that a successful programmer must use two primary tools: a good programming language (here it is C#) and design patterns.

When working on a problem, it is unusual to tackle it by inventing a new solution that is completely dissimilar from the existing ones. One often recalls a similar problem and reuses the essence of its solution to solve the new problem. This kind of thinking in problem-solving is common to many different domains, such as software engineering.

Design patterns are important building blocks for designing and modeling applications on all platforms. Design patterns help us understand, discuss, and reuse applications on a specific platform. The most common reasons for studying patterns are the reuse of solutions and the establishment of common terminology. By reusing established designs, a developer gets a headstart on the problem and avoids common mistakes. The benefit of learning from the experience of others' results is that the developer does not have to reinvent solutions for recurring problems. The other reason for using patterns is that common terminology brings a common base of vocabulary and viewpoint of the problem for developers. It provides a common point of reference during the analysis and design phase of a project.

Vaskaran Sarcar, who has worked with me for several years now, has been a Most Valuable Professional over the years in C#. He is enthusiastic, knowledgeable, talented, curious, analytical, and a teacher of others. He gets to the root of any problem he is trying to resolve in a well-defined and organized way. He is very committed and works hard until he gets to the solution. He gets involved and is deeply focused while working on any problem.

And that is also why I am excited about this book. The book brings the frequently complex world of design patterns into sharp focus with the approach used: the definition, the core concept, a real-life example, a computer-world example, and a sample program with output. In this edition, Vaskaran has provided asynchronous programming patterns usage using C#.

I look forward to seeing where developers can go with this easy approach and language, and the useful patterns they can build into the infrastructure of other languages.

—Priya Shimanthoor
Test Architect
Managed Print Services Team
Bangalore, India
June 3, 2020

Acknowledgments

First, I thank the Almighty. I sincerely believe that I could complete this book only with His blessings. I extend my deepest gratitude and thanks to the following people.

Ratanlal Sarkar and Manikuntala Sarkar: My dear parents, only with your blessings could I complete this work.

Indrani, my wife; **Ambika**, my daughter; **Aryaman**, my son: Sweethearts, once again, without your love, I could not proceed at all. I know that we need to limit many social gatherings and invitations to complete my books on time, and each time I promise you that I'll take a long break and spend more time with you.

Sambaran, my brother: Thank you for your constant encouragement.

Carsten: I know that whenever I was in need, your support was there. Thank you once more.

Sekhar: I know this time you helped only in the incremented version of the book, but thank you once more.

Ankit, my technical advisor in the first edition of this book: I always acknowledge your contribution and help. I know that your valuable comments were some of the key foundations for this enhanced edition.

Priya, my ex-colleague cum senior: A special thanks to you for investing your time to write the forewords for both editions of this book. When experts like you agree to write for me, I get the additional motivation to enhance the quality of my work.

Celestin, Laura, Smriti: Thanks for giving me another opportunity to work with you and Apress.

Shrikant: Thank you for your exceptional support to beautify my work.

The production team—Krishnan Sathyamurthy, Sherly, Ramraj, Selvakumar, and MathaRajamohan: Thank you guys; your efforts are extraordinary.

Lastly, I extend my deepest gratitude to my publisher, the editorial board members, and everyone who directly or indirectly supports this book.

Preface

Welcome to your journey through *Design Patterns in C# Second Edition*.

This book is an introductory guide to the design patterns that you want to use in C#. You probably know that the concept of design patterns became extremely popular with the Gang of Four's famous book *Design Patterns: Elements of Reusable Object-Oriented Software* (Addison-Wesley, 1994). That book was primarily focused on C++, but these concepts still apply in today's programming world.

C# had its first major release (C# 2.0) in 2005. Since then, it has become rich with new features and is now a popular programming language. In 2015, I wrote the book *Design Patterns in C#: Computer Science Interview Series*. In 2018, *Design Patterns in C#: A Hands-on Guide with Real-World Examples* was born. In these books, my core intention was to implement each of the 23 Gang of Four (GoF) design patterns with C# implementations. I wanted to present each pattern with simple examples. One thing was always on my mind when writing: I wanted to use the most basic constructs of C# so that the code would be compatible with both the upcoming version and the legacy version of C#. I have found this method helpful in the world of programming.

In the last few years, I got a lot of constructive feedback from my readers. This fully revised and updated version was created with that feedback in mind. I took the opportunity to update the formatting and correct some typos from the previous version of the book and add new content to this new edition. In this book, I focus on another important area; I call it the "doubt-clearing sessions." I knew that if I could add more information, such as alternative ways to write the implementations, the pros and cons of the patterns, when to choose one approach over another, and so on, readers would find this book even more helpful.

In this updated version of the original, the "Q&A Session" sections in each chapter are further enhanced. These sessions can help you learn about each pattern in more depth. In addition, you see more code explanations for all the programs, and in many cases, the programs are further simplified, and new programs are added for the patterns. *To learn about the most important enhancements in this edition, refer to* Appendix D at the end of this book.

How Is the Book Organized?

This book has three major parts.

Part I consists of the first 23 chapters, which discusses and implements all the GoF design patterns.

In the world of programming, there is no shortage of patterns, and each has its own significance. Part II discusses some additional design patterns (Simple Factory, Null Object, and MVC) that are equally important in today's world of programming. In this second edition, I dropped discussions of memory leaks, but I include several patterns from asynchronous programming. In modern applications, these patterns are very common.

Part III discusses the criticism of design patterns and overviews antipatterns, which are important when you implement the concepts of design patterns in your applications.

Each chapter is divided into six major parts: a definition (which is basically the intent in the GoF book), a core concept, a real-world example, a computer/coding–world example, a sample program with various output, and the "Q&A Session" section. These sections help you learn about each pattern in more depth.

Please remember that you have just started this journey. As you learn the concepts, try to write your own code; only then will you master an area.

You will be able to download all the book's source code from the Apress website. I plan to maintain the errata, and if necessary, I will also make updates and announcements there.

Prerequisite Knowledge

This book's target readers are those who are familiar with C# basic language constructs and pure object-oriented concepts, like polymorphism, inheritance, abstraction, encapsulation, and most importantly, how to compile or run a C# application in Visual Studio. This book does not invest time in easily available topics, such as how to install Visual Studio on your system, or how to write a "Hello World" program in C#, or how can you use an if-else statement or a while loop, and so forth. This book was written using the most basic features of C#, so for most of the programs in this book, you do not need to be familiar with C# advanced topics. The examples are simple and straightforward. I believe that the examples are written in such a way that even if you are familiar with another popular language, such as Java or C++, you can still easily grasp the concepts in this book.

Who Is This Book For?

In short, you want this book if your answer is "yes" to all of the following questions.

- Are you familiar with basic constructs in C# and object-oriented concepts like polymorphism, inheritance, abstraction, and encapsulation?
- Do you know how to set up your coding environment?
- Do you want to explore the design patterns in C# step by step?
- Do you want to explore GoF design patterns?
- Are you interested in learning about Simple Factory, Null Object, MVC, and asynchronous programming patterns?
- Do you want to know how the core constructs of C# work behind these patterns?

You probably don't want this book if the answer is "yes" to any of the following questions.

- Are you new to C#?
- Are you looking for advanced concepts in C#, excluding the topics mentioned previously?
- Are you interested in exploring a book where the focus is on GoF patterns (and the patterns listed in the previous section)?
- Do you dislike a book that uses Q&A sessions?
- "I do not like Windows, Visual Studio, and .NET Core. I want to learn and use C# without them." Is this statement true for you?

Guidelines for Using This Book

Here are some suggestions to help you use this book more effectively.

- I assume that you have some knowledge of GoF design patterns. If you are new to design patterns, I suggest you quickly go through Appendix A, which helps you become familiar with the basic concepts of design patterns.

- If you are confident with what Appendix A covers, you can start with any part of the book. But I suggest you go through the chapters sequentially. The reason is that some fundamental design techniques may have been discussed in the Q&A Session of a previous chapter, and I do not repeat those techniques in later chapters.
- There is only one exception to the previous suggestion. There are three factory patterns: Simple Factory, Factory Method, and Abstract Factory. These three patterns are closely related, but the Simple Factory pattern does not directly fall into the GoF design catalog, so it appears in Part II of the book. Therefore, of the three patterns, I suggest that you begin with the Simple Factory.
- Except for a few programs in Chapter 27, all programs were executed and tested in .NET Core 3.1. The remaining programs were executed in .NET Framework 4.7.2 because .NET Core doesn't support certain functionalities. The specific reasons are discussed in Chapter 27.
- I used Visual Studio Community edition 2019 (version 16.3.9) in a Windows 10 environment. This Community edition is free. If you do not use the Windows operating system, you can use Visual Studio Code, which is a source code editor developed by Microsoft to support Windows, Linux, or macOS operating systems. This multiplatform IDE is free. When I started writing this book, I used the latest versions of C# available. In this context, it is useful to know that the C# language version is automatically selected based on your project's target framework(s) so that you always have the highest compatible version by default. In the most recent versions, Visual Studio doesn't support the UI to change the value, but you can change it by editing the .csproj file. The Visual Studio 2019 compiler and the .NET Core 3.0 SDK follow this rule. Therefore, you can simply say that when your target framework is .NET Core 3.x (or newer), you'll get C# 8.0(and higher) by default. If you are interested in C# language versioning, go to `https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/configure-language-version`.

- Version updates are continuous, but I strongly believe that the versions should not matter much to you because I use the fundamental constructs of C# in this book. The code should execute smoothly in the upcoming versions of C#/Visual Studio as well. Although I believe that the results should not vary in other environments, you know the nature of a software-it can be naughty. So, I recommend that if you want to see the same output, it is best to mimic the same environment.
- You can download and install Visual Studio IDE from `https://visualstudio.microsoft.com/downloads/`. You should see what's shown in Figure P-1.

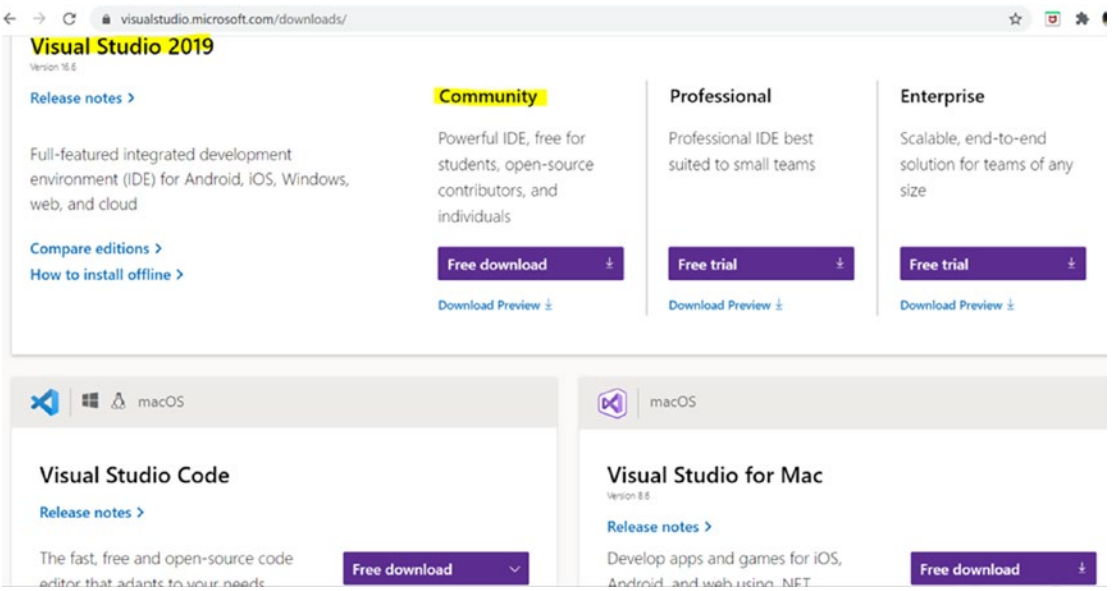

Figure P-1. *Download link for Visual Studio 2019 and Visual Studio Code*

Note At the time of this writing, this link works fine, and the information is correct. But the link and policies may change in the future.
