



# Design Patterns in C#

A Hands-on Guide with  
Real-World Examples

---

Vaskaran Sarcar

Foreword by Priya Shimanthoor

Apress®

# **Design Patterns in C#**

**A Hands-on Guide with  
Real-World Examples**

**Vaskaran Sarcar**

**Foreword by Priya Shimanthoor**

**Apress®**

## *Design Patterns in C#*

Vaskaran Sarcar  
Whitefield, Bangalore, Karnataka, India

ISBN-13 (pbk): 978-1-4842-3639-0  
<https://doi.org/10.1007/978-1-4842-3640-6>

ISBN-13 (electronic): 978-1-4842-3640-6

Library of Congress Control Number: 2018946636

Copyright © 2018 by Vaskaran Sarcar

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the author nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr  
Acquisitions Editor: Smriti Srivastava  
Development Editor: Matthew Moodie  
Coordinating Editor: Divya Modi

Cover designed by eStudioCalamar

Cover image designed by Freepik ([www.freepik.com](http://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](mailto:orders-ny@springer-sbm.com), or visit [www.springeronline.com](http://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 [rights@apress.com](mailto:rights@apress.com), or visit [www.apress.com/rights-permissions](http://www.apress.com/rights-permissions).

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 [www.apress.com/bulk-sales](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/978-1-4842-3639-0](http://www.apress.com/978-1-4842-3639-0). For more detailed information, please visit [www.apress.com/source-code](http://www.apress.com/source-code).

Printed on acid-free paper

*This book is dedicated to our second child who came into our lives for only two short months. He did not see the light of this world because he stopped growing with no significant heartbeat. We pray to Almighty God: "Wherever you are now, let His blessings be with you and make you happy. Sweetheart, though we could not give you life and reveal the beauty of this wonderful world to you, please remember that your parents always love you."*

# Table of Contents

**About the Author .....xix**

**About the Technical Reviewers .....xxi**

**Acknowledgments .....xxiii**

**Foreword .....xxv**

**Preface .....xxvii**

**Guidelines for Using This Book.....xxix**

**Part I: Gang of Four Design Patterns ..... 1**

**I.A: Creational Patterns..... 3**

**Chapter 1: Singleton Pattern ..... 5**

    GoF Definition..... 5

    Concept..... 5

    Real-Life Example ..... 5

    Computer World Example..... 5

    Illustration ..... 6

        Class Diagram ..... 6

        Solution Explorer View..... 7

        Discussion ..... 7

        Implementation ..... 8

        Output..... 9

        Challenges ..... 10

    Q&A Session..... 11

TABLE OF CONTENTS

**Chapter 2: Prototype Pattern ..... 17**

GoF Definition..... 17

Concept..... 17

Real-Life Example ..... 17

Computer World Example..... 17

Illustration ..... 18

    Class Diagram ..... 18

    Directed Graph Document ..... 20

    Solution Explorer View..... 20

    Implementation ..... 22

    Output..... 24

Q&A Session..... 25

    Demonstration ..... 27

    Output..... 29

**Chapter 3: Builder Pattern ..... 31**

GoF Definition..... 31

Concept..... 31

Real-Life Example ..... 32

Computer World Example..... 32

Illustration ..... 32

    Class Diagram ..... 33

    Solution Explorer View..... 33

    Implementation ..... 34

    Output..... 39

Q&A Session..... 39

**Chapter 4: Factory Method Pattern ..... 43**

GoF Definition..... 43

Concept..... 43

Real-Life Example ..... 43

|                                                 |           |
|-------------------------------------------------|-----------|
| Computer World Example.....                     | 44        |
| Illustration .....                              | 44        |
| Class Diagram .....                             | 44        |
| Directed Graph Document .....                   | 45        |
| Solution Explorer View.....                     | 45        |
| Implementation .....                            | 46        |
| Output.....                                     | 49        |
| Modified Implementation.....                    | 49        |
| Modified Output.....                            | 51        |
| Analysis .....                                  | 51        |
| Q&A Session.....                                | 51        |
| <b>Chapter 5: Abstract Factory Pattern.....</b> | <b>55</b> |
| GoF Definition.....                             | 55        |
| Concept.....                                    | 55        |
| Real-Life Example .....                         | 56        |
| Computer World Example.....                     | 56        |
| Illustration .....                              | 56        |
| Class Diagram .....                             | 58        |
| Solution Explorer View.....                     | 58        |
| Implementation .....                            | 60        |
| Output.....                                     | 63        |
| Q&A Session.....                                | 64        |
| Conclusion .....                                | 67        |
| <b>I.B: Structural Patterns.....</b>            | <b>69</b> |
| <b>Chapter 6: Proxy Pattern .....</b>           | <b>71</b> |
| GoF Definition.....                             | 71        |
| Concept.....                                    | 71        |
| Real-Life Example .....                         | 71        |
| Computer World Example.....                     | 71        |

TABLE OF CONTENTS

Illustration ..... 72

    Class Diagram ..... 72

    Directed Graph Document ..... 73

    Solution Explorer View..... 73

    Implementation ..... 74

    Output..... 76

Q&A Session..... 76

    Modified Implementation..... 79

    Modified Output..... 82

**Chapter 7: Decorator Pattern ..... 83**

    GoF Definition..... 83

    Concept..... 83

    Real-Life Example ..... 83

    Computer World Example..... 85

    Illustration ..... 85

        Class Diagram ..... 86

        Solution Explorer View..... 87

        Implementation ..... 88

        Output..... 90

    Q&A Session..... 91

**Chapter 8: Adapter Pattern ..... 97**

    GoF Definition..... 97

    Concept..... 97

    Real-Life Example ..... 97

    Computer World Example..... 98

    Illustration ..... 99

        Class Diagram ..... 99

        Directed Graph Document ..... 100

        Solution Explorer View..... 100

        Implementation ..... 102

|                                                         |            |
|---------------------------------------------------------|------------|
| Output.....                                             | 103        |
| Modified Illustration.....                              | 103        |
| Key Characteristics of the Modified Implementation..... | 104        |
| Modified Solution Explorer View.....                    | 105        |
| Modified Implementation.....                            | 106        |
| Modified Output.....                                    | 109        |
| Types of Adapters.....                                  | 109        |
| Q&A Session.....                                        | 110        |
| <b>Chapter 9: Facade Pattern .....</b>                  | <b>113</b> |
| GoF Definition.....                                     | 113        |
| Concept.....                                            | 113        |
| Real-Life Example .....                                 | 113        |
| Computer World Example.....                             | 113        |
| Illustration .....                                      | 114        |
| Class Diagram .....                                     | 114        |
| Directed Graph Document .....                           | 115        |
| Solution Explorer View.....                             | 115        |
| Implementation .....                                    | 116        |
| Output.....                                             | 120        |
| Q&A Session.....                                        | 121        |
| <b>Chapter 10: Flyweight Pattern .....</b>              | <b>123</b> |
| GoF Definition.....                                     | 123        |
| Concept.....                                            | 123        |
| Real-Life Example .....                                 | 124        |
| Computer World Example.....                             | 124        |
| Illustration .....                                      | 124        |
| Class Diagram .....                                     | 125        |
| Directed Graph Document.....                            | 126        |
| Solution Explorer View .....                            | 127        |

TABLE OF CONTENTS

|                                           |            |
|-------------------------------------------|------------|
| Implementation .....                      | 127        |
| Output .....                              | 130        |
| Improvements to the Program .....         | 131        |
| Modified Class Diagram .....              | 132        |
| Modified Solution Explorer View .....     | 133        |
| Modified Implementation .....             | 133        |
| Modified Output .....                     | 137        |
| Q&A Session.....                          | 138        |
| <b>Chapter 11: Composite Pattern.....</b> | <b>143</b> |
| GoF Definition.....                       | 143        |
| Concept.....                              | 143        |
| Real-Life Example .....                   | 144        |
| Computer World Example.....               | 144        |
| Illustration .....                        | 144        |
| Class Diagram .....                       | 146        |
| Solution Explorer View.....               | 147        |
| Implementation .....                      | 148        |
| Output.....                               | 151        |
| Q&A Session.....                          | 152        |
| <b>Chapter 12: Bridge Pattern .....</b>   | <b>155</b> |
| GoF Definition.....                       | 155        |
| Concept.....                              | 155        |
| Real-Life Example .....                   | 155        |
| Computer World Example.....               | 156        |
| Illustration .....                        | 156        |
| Class Diagram .....                       | 158        |
| Solution Explorer View.....               | 159        |
| Implementation .....                      | 159        |
| Output.....                               | 163        |
| Q&A Session.....                          | 163        |

|                                           |            |
|-------------------------------------------|------------|
| <b>I.C: Behavioral Patterns .....</b>     | <b>165</b> |
| <b>Chapter 13: Visitor Pattern .....</b>  | <b>167</b> |
| GoF Definition.....                       | 167        |
| Concept.....                              | 167        |
| Real-Life Example .....                   | 167        |
| Computer World Example.....               | 168        |
| Illustration .....                        | 168        |
| Class Diagram .....                       | 169        |
| Solution Explorer View.....               | 170        |
| Implementation .....                      | 170        |
| Output.....                               | 172        |
| Modified Illustration.....                | 172        |
| Modified Solution Explorer View .....     | 176        |
| Modified Implementation.....              | 177        |
| Modified Output.....                      | 183        |
| Q&A Session.....                          | 184        |
| <b>Chapter 14: Observer Pattern .....</b> | <b>185</b> |
| GoF Definition.....                       | 185        |
| Concept.....                              | 185        |
| Real-Life Example .....                   | 188        |
| Computer World Example.....               | 188        |
| Illustration .....                        | 189        |
| Class Diagram .....                       | 189        |
| Solution Explorer View.....               | 190        |
| Implementation .....                      | 191        |
| Output.....                               | 194        |
| Analysis of the Output .....              | 194        |
| Q&A Session.....                          | 195        |

TABLE OF CONTENTS

**Chapter 15: Strategy (Policy) Pattern ..... 199**

GoF Definition..... 199

Concept..... 199

Real-Life Example ..... 199

Computer World Example..... 199

Illustration ..... 200

    Class Diagram ..... 200

Directed Graph Document..... 201

Solution Explorer View ..... 202

Implementation ..... 203

Output ..... 205

Q&A Session..... 206

**Chapter 16: Template Method Pattern ..... 211**

GoF Definition..... 211

Concept..... 211

Real-Life Example ..... 211

Computer World Example..... 212

Illustration ..... 212

    Class Diagram ..... 213

    Solution Explorer View..... 214

    Implementation ..... 214

    Output..... 217

Q&A Session..... 217

    Modified Implementation..... 218

    Modified Output..... 221

**Chapter 17: Command Pattern ..... 223**

GoF Definition..... 223

Concept..... 223

Real-Life Example ..... 223

Computer World Example..... 224

|                                           |            |
|-------------------------------------------|------------|
| Illustration .....                        | 224        |
| Class Diagram .....                       | 225        |
| Directed Graph Document .....             | 226        |
| Solution Explorer View .....              | 227        |
| Implementation .....                      | 228        |
| Output .....                              | 231        |
| Q&A Session .....                         | 231        |
| Modified Class Diagram .....              | 232        |
| Modified Solution Explorer View .....     | 233        |
| Modified Implementation .....             | 234        |
| Modified Output .....                     | 240        |
| <b>Chapter 18: Iterator Pattern .....</b> | <b>243</b> |
| GoF Definition .....                      | 243        |
| Concept .....                             | 243        |
| Real-Life Example .....                   | 244        |
| Computer World Example .....              | 244        |
| Illustration .....                        | 244        |
| Class Diagram .....                       | 245        |
| Directed Graph Document .....             | 246        |
| Solution Explorer View .....              | 247        |
| Implementation .....                      | 248        |
| Output .....                              | 253        |
| Q&A Session .....                         | 253        |
| <b>Chapter 19: Memento Pattern .....</b>  | <b>257</b> |
| GoF Definition .....                      | 257        |
| Concept .....                             | 257        |
| Real-Life Example .....                   | 257        |
| Computer World Example .....              | 257        |

TABLE OF CONTENTS

Illustration ..... 258

    Class Diagram ..... 258

    Directed Graph Document ..... 259

    Solution Explorer View ..... 260

    Implementation ..... 260

    Output ..... 264

Q&A Session ..... 264

    Modified Implementation ..... 264

    Modified Output ..... 266

**Chapter 20: State Pattern ..... 269**

    GoF Definition ..... 269

    Concept ..... 269

    Real-Life Example ..... 269

    Computer World Example ..... 270

    Illustration ..... 270

        Class Diagram ..... 271

        Directed Graph Document ..... 272

        Solution Explorer View ..... 272

        Implementation ..... 274

        Output ..... 279

    Q&A Session ..... 279

**Chapter 21: Mediator Pattern ..... 283**

    GoF Definition ..... 283

    Concept ..... 283

    Real-Life Example ..... 283

    Computer World Example ..... 284

    Illustration ..... 284

        Class Diagram ..... 286

        Solution Explorer View ..... 286

        Implementation ..... 288

|                                                         |            |
|---------------------------------------------------------|------------|
| Output.....                                             | 292        |
| Analysis .....                                          | 292        |
| Q&A Session.....                                        | 293        |
| Modified Illustration.....                              | 294        |
| Modified Implementation.....                            | 294        |
| Modified Output.....                                    | 299        |
| <b>Chapter 22: Chain of Responsibility Pattern.....</b> | <b>303</b> |
| GoF Definition.....                                     | 303        |
| Concept.....                                            | 303        |
| Real-Life Example .....                                 | 304        |
| Computer World Example.....                             | 304        |
| Illustration .....                                      | 305        |
| Class Diagram .....                                     | 306        |
| Directed Graph Document .....                           | 307        |
| Solution Explorer View.....                             | 307        |
| Implementation .....                                    | 308        |
| Output.....                                             | 312        |
| Q&A Session.....                                        | 312        |
| <b>Chapter 23: Interpreter Pattern .....</b>            | <b>315</b> |
| GoF Definition.....                                     | 315        |
| Concept.....                                            | 315        |
| Real-Life Example .....                                 | 315        |
| Computer World Example.....                             | 315        |
| Illustration .....                                      | 316        |
| Class Diagram .....                                     | 316        |
| Solution Explorer View.....                             | 318        |
| Implementation .....                                    | 318        |
| Output.....                                             | 325        |
| Q&A Session.....                                        | 326        |

**Part II: Additional Design Patterns ..... 327**

**Chapter 24: Simple Factory Pattern ..... 329**

    Definition..... 329

    Concept..... 329

    Real-Life Example ..... 329

    Computer World Example..... 330

    Illustration ..... 330

        Class Diagram ..... 331

        Directed Graph Document ..... 332

        Solution Explorer View..... 332

        Implementation ..... 333

        Output..... 336

    Q&A Session..... 337

**Chapter 25: Null Object Pattern ..... 341**

    Definition..... 341

    Concept..... 341

        A Faulty Program ..... 342

        Output with Valid Inputs ..... 344

        Analysis with Unwanted Input ..... 344

        Encountered Exception..... 345

        Immediate Remedy..... 345

        Analysis ..... 345

    Real-Life Example ..... 346

    Computer World Example..... 346

    Illustration ..... 346

        Class Diagram ..... 347

        Solution Explorer View..... 348

        Implementation ..... 349

        Output..... 352

    Q&A Session..... 353

|                                                                |            |
|----------------------------------------------------------------|------------|
| <b>Chapter 26: MVC Pattern .....</b>                           | <b>355</b> |
| Definition.....                                                | 355        |
| Concept.....                                                   | 355        |
| Key Points to Remember .....                                   | 356        |
| Variation 1 .....                                              | 357        |
| Variation 2 .....                                              | 358        |
| Variation 3 .....                                              | 358        |
| Real-Life Example .....                                        | 359        |
| Computer World Example.....                                    | 359        |
| Illustration .....                                             | 360        |
| Class Diagram .....                                            | 362        |
| Solution Explorer View.....                                    | 362        |
| Implementation .....                                           | 363        |
| Output.....                                                    | 373        |
| Q&A Session.....                                               | 377        |
| <b>Part III: Final Thoughts on Design Patterns .....</b>       | <b>383</b> |
| <b>Chapter 27: Criticisms of Design Patterns.....</b>          | <b>385</b> |
| Q&A Session.....                                               | 387        |
| <b>Chapter 28: Anti-patterns.....</b>                          | <b>391</b> |
| What Is an Anti-pattern? .....                                 | 391        |
| Q&A Session.....                                               | 392        |
| <b>Chapter 29: Sealing the Leaks in Your Applications.....</b> | <b>397</b> |
| How Garbage Collection Works .....                             | 398        |
| Demonstration 1.....                                           | 400        |
| Output.....                                                    | 403        |
| Analysis .....                                                 | 404        |
| Q&A Session 1.....                                             | 405        |
| Quiz.....                                                      | 413        |
| Quiz.....                                                      | 415        |

TABLE OF CONTENTS

Quiz..... 417

Quiz..... 418

Understanding Memory Leaks ..... 419

Demonstration 2..... 420

    Snapshots..... 422

    Analysis ..... 423

    Modified Code..... 424

    Snapshots..... 427

    Analysis ..... 428

Q&A Session 2..... 430

**Chapter 30: FAQ ..... 433**

**Appendix A: Brief Overview of GoF Design Patterns..... 439**

    Key Points ..... 440

    Q&A Session..... 443

**Appendix B: Some Useful Resources ..... 445**

**Appendix C: The Road Ahead ..... 447**

**Index..... 449**

# About the Author



**Vaskaran Sarcar** obtained his Master of Engineering degree from Jadavpur University, Kolkata in Software Engineering. Currently he is a Senior Software Engineer and Team Lead in the R&D Hub at HP Inc. India. He was a national Gate Scholar and has more than 12 years of experience in education and the IT industry. He is an alumnus of prestigious institutions in India such as Jadavpur University, Vidyasagar University, and Presidency University (formerly Presidency College). Reading and learning new things are his passions. You can connect with him at [vaskaran@rediffmail.com](mailto:vaskaran@rediffmail.com) or find him on LinkedIn at

<https://www.linkedin.com/in/vaskaransarcar>.

Other books by Vaskaran include the following:

*Interactive C#* (Apress, 2017)

*Interactive Object-Oriented Programming in Java* (Apress, 2016)

*Java Design Patterns* (Apress, 2016)

*C# Basics: Test Your Skill* (CreateSpace, 2015)

*Operating System: Computer Science Interview Series* (CreateSpace, 2014)

# About the Technical Reviewers



**Shekhar Kumar Maravi** is a system software engineer 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 HP's R&D Hub in India to work on printer firmware. Currently he is a technical lead for automated lab diagnostic device firmware and software at Siemens Healthcare India. He can be reached by e-mail at [shekhar.maravi@gmail.com](mailto:shekhar.maravi@gmail.com) or via LinkedIn at <https://www.linkedin.com/in/shekharmaravi>.



**Ankit Khare** is a senior software engineer with expertise in software architecture and designing, programming languages, algorithms, and data structure. After obtaining a bachelor's degree in Computer Science, he joined HP's R&D Hub in India, where he worked with various laser-jet firmware teams. He is currently involved in future machine vision development for print image diagnostic tools involving ink-jet, large-format, and laser-jet printers. He can be reached by e-mail at [akikhare@gmail.com](mailto:akikhare@gmail.com) or at <https://www.linkedin.com/in/khareankit/>.

# Acknowledgments

First, I thank the Almighty. I sincerely believe that only with His blessings I was able to complete this book.

I extend my deepest gratitude and thanks to the following people:

Ratanlal Sarkar and Manikuntala Sarkar, my parents. Without your blessings, I could not have completed this work.

Indrani, my wife, and Ambika, my daughter. Sweethearts, once again, without your love, I could not proceed at all. I know that we needed to limit many social gatherings and invitations to complete this work on time.

Sambaran, my brother. Thank you for your constant encouragement toward me.

Shekhar and Ankit, my friends and technical advisers. Whenever I was in need, your support was there. Thank you one more time.

Priya Shimanthoor, my colleague and mentor. A special thanks to you for investing your time to write a foreword for my book. From the moment that an expert like you agreed to write about me, I was motivated to enhance the quality of my work.

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

# Foreword



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

When working on a particular problem, it is unusual to tackle it by inventing a new solution that is completely dissimilar from existing ones. Instead, one often recalls a similar problem and reuses the essence of its solution to solve the new problem. This kind of problem-solving is common to many different domains but especially 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 commonly stated reasons for studying patterns are to reuse solutions and to establish common terminology. By reusing already established designs, the developer gets a head start on the problem and avoids common mistakes. The benefit of learning from the experience of others results in not having to reinvent solutions for commonly recurring problems. The other reason for using patterns is that common terminology brings a shared base of vocabulary and viewpoint of the problem to the developers. It provides a point of reference during the analysis and design phases of a project.

Vaskaran Sarcar, who has worked with me for several years now, has been our C# team's most valuable professional over the years. Vaskaran is that kind of software developer—enthusiastic, knowledgeable, talented, curious, analytical, and a teacher of others. He gets into the root of any particular problem he is trying to solve in a well-defined and organized way. He is committed and works hard until he gets to the solution.

## FOREWORD

That is why I am excited about this book. *Design Patterns in C#* brings the frequently abstruse world of design patterns into sharp focus with the approach used: a definition, a core concept, a real-life example, a computer world example, and a sample program with output. I look forward to see where developers can go with this easy approach and language and what useful patterns they can build into the infrastructure of other languages.

Priya Shimanthoor  
Test Architect, Firmware Team  
HP India PPS R&D Hub  
Bangalore, India  
April 8, 2018

### About Priya Shimanthoor

Priya Shimanthoor completed her graduation with Computer Science and Engineering in 2000. Currently she is a Test Architect. In last 17 years, she tested different Desktop/web applications, software solutions, firmware and mobile applications using various tools like WinRunner, Quick Test Professional, Rational Robot, Rational Test Manager, LoadRunner, Test Director, Quality Center, Application lifecycle management, Python Everest Framework and JUnit. She also worked on various virtualization solutions and she is well-versed with different process models like CMMi, Agile, Lean Sigma etc.

# Preface

Welcome to your journey of *Design Patterns 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). Most important, these concepts still apply in today's programming world. The book came out at the end of 1994, and it primarily focused on C++. In 2005, C# had its first major release (C# 2.0). Since then, C# has become rich with features and is now a popular programming language.

In 2015, I wrote the book *Design Patterns in C#: Computer Science Interview Series*, which was basically a companion to this book.

In that tiny book, 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 in 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 two years, I have added enhancements to the book based on readers feedback. In 2017, Visual Studio launched with the upgraded version of C#, and it gave me pleasure when I saw the code being executed as expected with the updated version of C#.

This time, I wanted to focus on another important area; I call it the “doubt-clearing sessions.” I knew that if I could add some more information such as alternative ways to write these implementations, the pros and cons of these patterns, when to choose one approach over another, and so on, readers would find this book even more helpful. So, in this enhanced version of the original, I have added a “Q&A Session” section to each chapter that can help you to learn about each pattern in more depth.

In the world of programming, there is no shortage of patterns, and each has its own significance. So, in addition to the 23 GoF design patterns covered in Part I, I discuss three design patterns in Part II of the book that are equally important in today's world of programming. Finally, in Part III of the book, I give you an overview of anti-patterns and discuss the criticism of design patterns with regard to one important concept called *memory leak*. I have included this topic because in software development we cannot stamp the word *good* on the implementation of a design if it suffers from memory leaks.

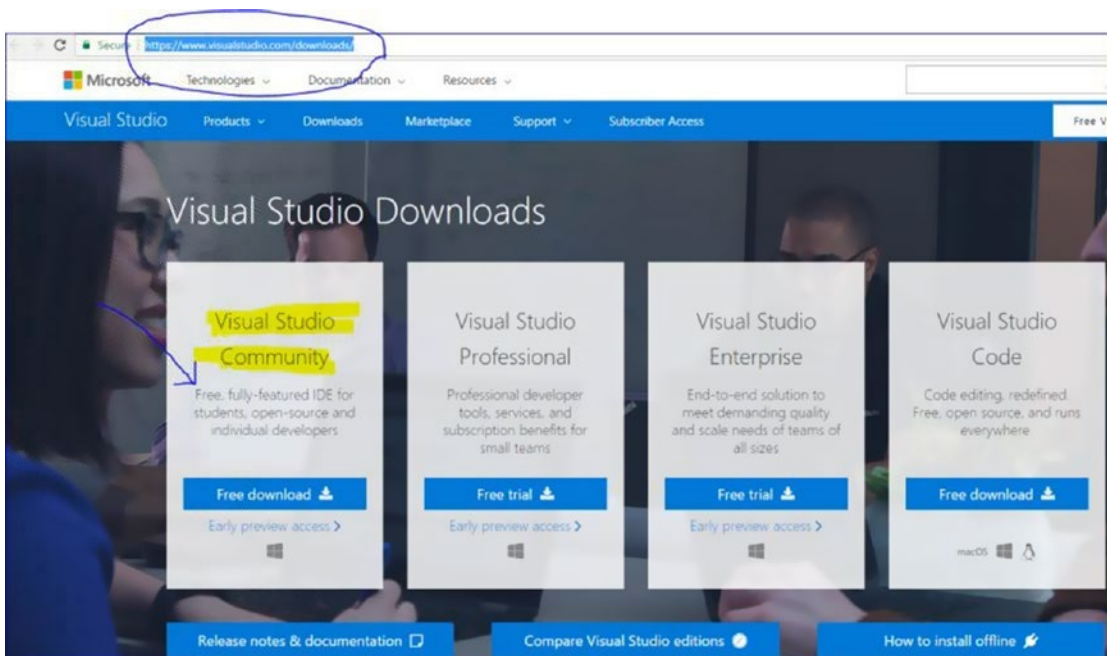
Before jumping into the design patterns, I want to highlight a few more points.

- You are an intelligent person. You have chosen a subject that can assist you throughout your career. If you are a developer/programmer, you need these concepts. If you are an architect of a software organization, you need these concepts. If you are a college student, you need these concepts, not only to score high on exams but to enter the corporate world. Even if you are a tester who needs to take care of white-box testing or simply needs to know about the code paths of a product, these concepts will help you a lot.
- I already mentioned that this book was written using the most basic features of C# so that you do not need to be familiar with advanced topics in C#. These examples are simple and straightforward. I believe that these examples are written in such a way that even if you are familiar with another popular language such as Java, C++, and so on, you can still easily grasp the concepts in this book.
- There are many books about design patterns and related topics. You may be wondering why I would want to write a new one about the same topics. The simple answer to this question is that I have found other reference material to be scattered. Second, in most cases, many of those examples are unnecessarily large and complex. I like simple examples. I believe that anyone can grasp a new idea with simple examples, and if the core concept is clear, you can easily move into more advanced areas. I believe that this book scores high in this context. The illustrated examples are simple. I wanted to keep this book concise so that it motivates you to continue the journey of learning in your field.
- Each chapter is divided into six parts: a definition (which is basically termed as intent in GoF book), a core concept, a real-life example, a computer/coding world example, a sample program with various outputs, and the “Q&A Session” section. These “Q&A Session” sections can help you to learn about each pattern in more depth.
- Please remember that you have just started on this journey. As you learn about these concepts, try to write your own code; only then will you master this area.

# Guidelines for Using This Book

Here are some suggestions so you can use the book more effectively:

- I assume that you have some idea about the GoF design patterns. If you are absolutely new to design patterns, I suggest you quickly go through Appendix A. This appendix will help you to become familiar with the basic concepts of design patterns.
- If you are confident with the coverage of Appendix A, 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 be discussed in the “Q&A Session” section of a previous chapter, and I have not repeated those techniques in the 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 related, but the Simple Factory pattern does not directly fall into the GoF design catalog, so it appears in Part II of the book. So, it is suggested that when you start learning about these 3 factory patterns, you start with the Simple Factory Pattern.
- These programs are tested with Visual Studio Community edition (2017) in a windows 10 environment. This Community edition is free of cost. If you do not use windows operating system, you can use Visual Studio Code Development environment IDE that can support Linux or Mac operating system. This multi-platform IDE is also free. Though I believe that the results should not vary in other environments, but you know the nature of softwares-they are naughty. So, I recommend that if you want to see the exact same outputs, it will be better if you can mimic the same environment.
- You can download and install the Visual Studio IDE from <https://www.visualstudio.com/downloads/> (Figure 1). You can use the free Visual Studio Community edition.



**Figure 1.** Download link for Visual Studio

---

**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.

---

- UML designers are removed in Visual Studio 2017. To know the reason, you can refer the following link:

<https://blogs.msdn.microsoft.com/devops/2016/10/14/uml-designers-have-been-removed-layer-designer-now-supports-live-architectural-analysis/>.

In this book, I have drawn few Directed Graph Document (DAG) to explain the things better. To serve those needs, as per the recommendation, I have used an older version of Visual Studio (Ultimate 2013) to draw those. But to understand the concepts, these are not mandatory for you. So, you can move smoothly with Visual Studio Community 2017.

# **PART I**

## **Gang of Four Design Patterns**

# **I.A: Creational Patterns**

