

The
Pragmatic
Programmers

C# Brain Teasers

Exercise Your Mind



Steve Love

edited by Adaobi Obi Tulton

This extract shows the online version of this title, and may contain features (such as hyperlinks and colors) that are not available in the print version.

For more information, or to purchase a paperback or ebook copy, please visit <https://www.pragprog.com>.

Copyright © The Pragmatic Programmers, LLC.

Introduction

Every useful programming language has its fair share of complexity, and C# is no different. Complexity brings the potential for hazards that can trip up the unwary programmer, causing problems from inexplicable compilation failures to errors in the output or even programs exiting unexpectedly.

C# is an evolving language, and each new feature adds its own new complexity. Whether it's one of the big-ticket items like LINQ or `async`, or a smaller feature such as primary constructors or field initializers, features can still introduce new hazards to avoid. Some of the puzzles tackle issues introduced in more recently added features of C#, while others address long-standing pitfalls that have caused programmers' headaches for years.

The first step in avoiding those pitfalls is to be able to recognize them when you encounter them. The puzzles in this book illustrate some common problems so you'll be able to identify them in your own code or the code you're working on.

About the Puzzles

Each puzzle starts with a simple program and asks you to determine what you think the output should be. The following discussion shows the actual output where possible; some of the puzzles won't compile successfully. The discussion goes on to explain *why* the result happens and also *how* to avoid the hazard being demonstrated.

Of course a book such as this can't possibly attempt to investigate all the dusty corners and subtleties of C#. Instead, I've tried to distil the puzzles down to those that demonstrate the everyday mistakes made by C# programmers—myself included!—in the course of writing C# programs, whether that's for work or as a hobby. Nevertheless, whether you're just starting out on your C# journey or you're an experienced C# programmer, I hope that at least some of the puzzles surprise you as we look into loops, asynchronicity, exceptions, numbers, and more. Moreover, I hope you'll be inspired to look

closely at other aspects of C# not covered here and learn more about why C# is the language it has become.

About You

The book assumes you have some experience with writing C# programs—for instance, that you're familiar with the basic structure of a C# program and the essentials of writing your own classes with properties and methods. The book also assumes you're familiar with compiling and running a C# program, but there are plenty of resources online if you need any help there.

Compiling and Running the Code

If you want to compile and run all the examples, I recommend you get .NET v8.0 or later.¹ Some of the code depends on features introduced in that version. It's freely available and comes with everything you need to get started with building C# programs from a shell or command prompt.

If you're using .NET Framework (the legacy nonportable framework and compiler) you'll need to do some extra work to get the code to compile. Again there are online resources that will help with that. Additionally, .NET Framework produces *different* results to .NET in some cases, and those are noted explicitly in the relevant puzzle discussions.

The concept of *top-level statements* was introduced in .NET v6.0. They do away with the need to have a separate class with a static Main method for small programs. Accordingly the code for each puzzle is structured so you can copy it directly into a file and run it in a console.

You can download the code for each puzzle from the book's web page,² where you can also join in discussions with other readers and report errata. If you're reading the ebook, clicking the filename that precedes each puzzle's code will download the code for that puzzle.

That's it—you're all set! Let's get on with it.

1. <https://dotnet.microsoft.com/download>
2. <http://pragprog.com/titles/csharpbt>