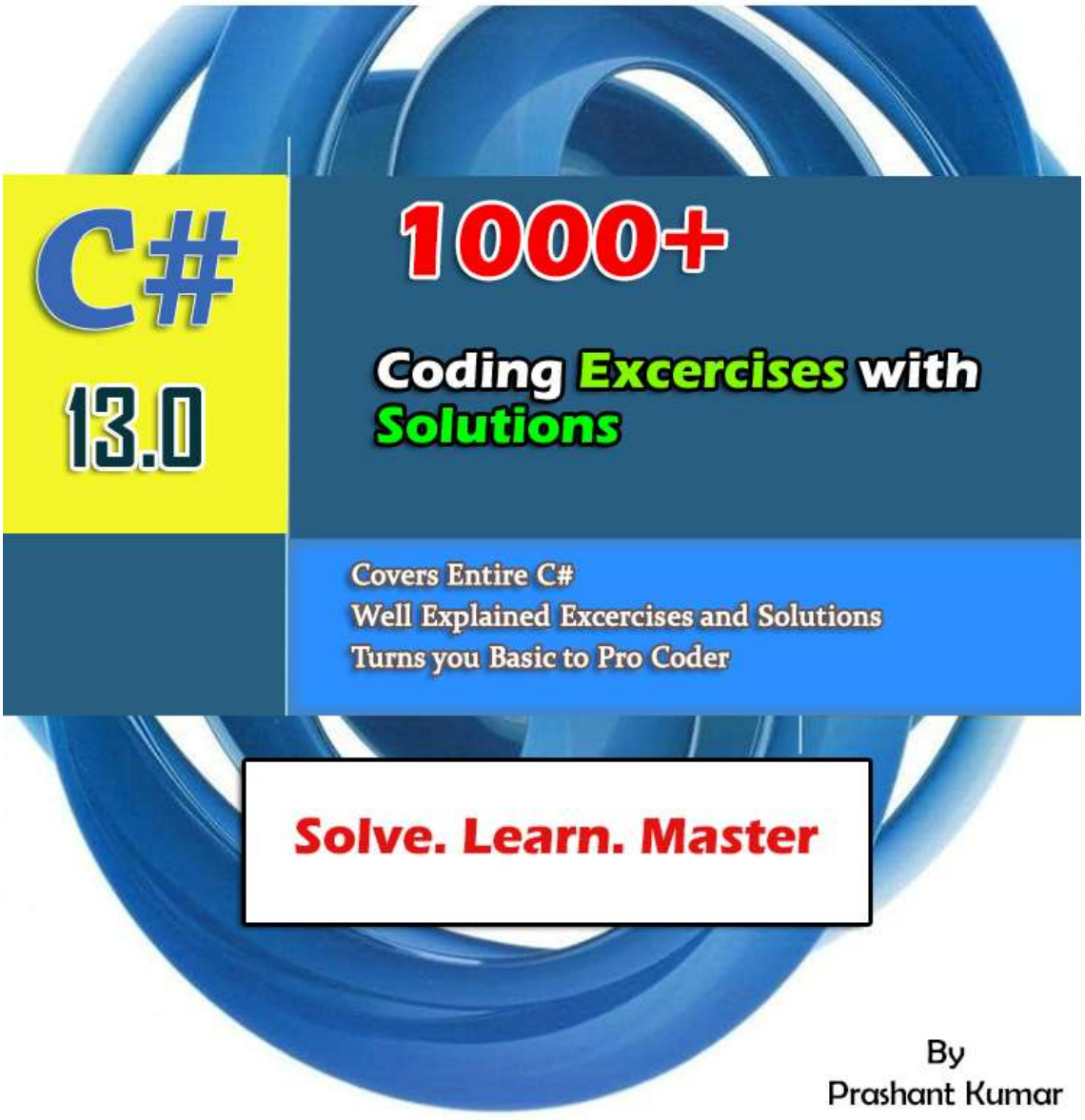
Your Ultimate Guide to Mastering C# with Practical Challenges and Expert Answers



By
Prashant Kumar

Your Ultimate Guide to Mastering C# with Practical Challenges and Expert Answers

Master Coding: 100+ Exercises with Solutions for Programmers

By Prashant Kumar

Book Content

1. **Control Flow Statements**
   - if
   - else
   - switch
   - case
   - default
   - for
   - foreach
   - while
   - do
   - break
   - continue
   - return
   - goto

2. **Data Types**
   - bool
   - byte
   - sbyte
   - char
   - decimal
   - double
   - float
   - int
   - uint
   - long
   - ulong
   - short

- ushort
- string
- object
- dynamic

3. **Access Modifiers**
   - public
   - private
   - protected
   - internal
   - protected internal
   - private protected

4. **Exception Handling**
   - try
   - catch
   - finally
   - throw

5. **Value Modifiers**
   - const
   - readonly
   - volatile

6. **Method Modifiers**
   - abstract
   - virtual
   - override
   - static
   - sealed
   - async

7. **Class Modifiers**
   - abstract
   - sealed
   - partial

8. **Namespace and Assembly**
   - namespace
   - using
   - extern

9. **Object-Oriented Programming**
   - class
   - struct

- interface
- enum
- delegate
- this
- base
- new

10. Generics
- where
- in
- out

11. Operators
- is
- as
- sizeof
- typeof
- checked
- unchecked

12. Other Contextual Keywords
- null
- true
- false
- var
- dynamic
- default

13. Unsafe Code
- unsafe
- fixed
- stackalloc

14. Threading
- lock
- yield

15. Keywords Related to Assemblies
- ref
- out
- params

16. Preprocessor Directives
- #if
- #else
- #elif

- #endif
- #define
- #undef
- #warning
- #error
- #line
- #region
- #endregion
- #pragma

17. Query Keywords (LINQ)

- from
- where
- select
- group
- into
- orderby
- join
- let
- in
- on
- equals

18. Asynchronous Programming

- async
- await

19. Special Context Keywords

- _ (discard variable in newer versions of C#)
- value (used in properties)

How to Approach These Problems:

1. Understand the Problem Statement: Clearly define what the input is, what output is expected, and what conditions must be met.
2. Choose the Right Loop: Use for, foreach, or while loops depending on the requirements.
3. Use Built-in Methods: Utilize .NET library functions like Array.Sort, Array.Reverse, etc., when applicable.
4. Debug and Test: Test your program with edge cases (e.g., empty arrays, arrays with all identical elements, etc.).

1. Control Flow Statement

1. IF Statement

Challenge 1: Check If a Number is Positive or Negative

Question:
Write a program that asks the user to enter a number and checks if it is positive, negative, or zero. Print an appropriate message for each case.

Hints:

1. Use if for checking if the number is positive or negative.
2. Use else for zero.

Code:

```
using System;

class Program
{
    static void Main()
    {
        Console.Write("Enter a number: ");
        int number = int.Parse(Console.ReadLine());

        if (number > 0)
        {
            Console.WriteLine("The number is positive.");
        }
        else if (number < 0)
        {
            Console.WriteLine("The number is negative.");
        }
```

```
        else
        {
            Console.WriteLine("The number is zero.");
        }
    }
}
```

Expected Output:

- Input: 5
  Output: The number is positive.
- Input: -3
  Output: The number is negative.
- Input: 0
  Output: The number is zero.

Challenge 2: Check Even or Odd

Question:
Write a program that asks the user to enter a number and checks if it is even or odd.

Hints:

1. Use the modulus operator % to check if the number is divisible by 2.
2. Use if for even and else for odd.

Code:

```
using System;

class Program
{
    static void Main()
    {
        Console.Write("Enter a number: ");
        int number = int.Parse(Console.ReadLine());

        if (number % 2 == 0)
        {
            Console.WriteLine("The number is even.");
        }
        else
        {
            Console.WriteLine("The number is odd.");
        }
    }
}
```

Expected Output:

- Input: 4
  Output: The number is even.

- Input: 7
  Output: The number is odd.

Challenge 3: Grade Based on Marks

Question:
Write a program that assigns a grade based on the marks entered by the user. Use the following scale:

- 90-100: A
- 80-89: B
- 70-79: C
- 60-69: D
- Below 60: F

Hints:

1. Use multiple if-else if conditions to check the range of marks.

Code:

```
using System;

class Program
{
    static void Main()
    {
        Console.Write("Enter your marks: ");
        int marks = int.Parse(Console.ReadLine());

        if (marks >= 90 && marks <= 100)
        {
            Console.WriteLine("Grade: A");
        }
        else if (marks >= 80 && marks < 90)
        {
            Console.WriteLine("Grade: B");
        }
        else if (marks >= 70 && marks < 80)
        {
            Console.WriteLine("Grade: C");
        }
        else if (marks >= 60 && marks < 70)
        {
            Console.WriteLine("Grade: D");
        }
        else
        {
            Console.WriteLine("Grade: F");
        }
    }
}
```

Expected Output:

- Input: 85
  Output: Grade: B
- Input: 45
  Output: Grade: F

Challenge 4: Check Leap Year

Question:
Write a program that asks the user to enter a year and checks if it is a leap year.

Hints:

1. A year is a leap year if it is divisible by 4 but not divisible by 100, except if it is also divisible by 400.
2. Use nested if conditions to implement the logic.

Code:

```
using System;

class Program
{
    static void Main()
    {
        Console.Write("Enter a year: ");
        int year = int.Parse(Console.ReadLine());

        if (year % 4 == 0)
        {
            if (year % 100 == 0)
            {
                if (year % 400 == 0)
                {
                    Console.WriteLine($"{year} is a leap year.");
                }
                else
                {
                    Console.WriteLine($"{year} is not a leap year.");
                }
            }
            else
            {
                Console.WriteLine($"{year} is a leap year.");
            }
        }
        else
        {
            Console.WriteLine($"{year} is not a leap year.");
        }
    }
}
```

Expected Output:

- Input: 2020
  Output: 2020 is a leap year.
- Input: 1900
  Output: 1900 is not a leap year.
- Input: 2000
  Output: 2000 is a leap year.

Challenge 5: Authenticate User

Question:
Write a program that checks if the entered username and password match predefined values (admin and 12345). Display a message indicating whether authentication was successful or not.

Hints:

1. Use if to compare the entered values with the predefined ones.
2. Combine conditions with && for checking both username and password.

Code:

```
using System;

class Program
{
    static void Main()
    {
        string predefinedUsername = "admin";
        string predefinedPassword = "12345";

        Console.Write("Enter username: ");
        string username = Console.ReadLine();

        Console.Write("Enter password: ");
        string password = Console.ReadLine();

        if (username == predefinedUsername && password == predefinedPassword)
        {
            Console.WriteLine("Authentication successful.");
        }
        else
        {
            Console.WriteLine("Authentication failed.");
        }
    }
}
```

Expected Output:

- Input:
  Username: admin

Password: 12345
Output: Authentication successful.

- Input:
  Username: user
  Password: password
  Output: Authentication failed.