

INetTutor.com

# Online Programming Lessons, Tutorials and Capstone Project guide

## Divisibility Checker in CSharp

Welcome to our C# programming tutorial on the Divisibility Checker! In this lesson, we'll explore a simple yet powerful program that helps us figure out if one number can be evenly divided by another. Plus, you can download the complete source code and a handy PDF tutorial. Let's dive in and discover the world of divisibility in C# programming!

### Introduction

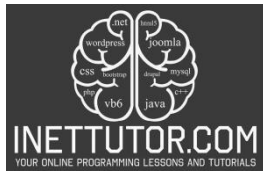
#### A. Explanation of the Program's Purpose

- The program's goal is to determine whether a given number is divisible by another number and to illustrate the concept of using the modulo operator for this purpose. B. Importance of Divisibility Checking
- Divisibility checks are crucial in various mathematical and programming scenarios. They help identify factors, multiples, and patterns in numbers, which is essential for tasks like prime number determination, finding common divisors, and optimizing algorithms.

In this lesson, we'll explore how to create a C# console program that checks for divisibility between two numbers and uses the modulo operator to simplify the process. Understanding divisibility is fundamental for solving mathematical problems and optimizing algorithms in programming.

### Objectives

1. Learn: Gain a solid understanding of the divisibility concept and how to implement it effectively in C# programming.
2. Apply: Apply divisibility checking techniques to solve real-world problems across various domains, reinforcing practical problem-solving skills.
3. Create: Develop a fully functional C# program for divisibility checking, demonstrating proficiency in algorithmic thinking and code implementation.
4. Upgrade: Elevate your programming skills by enhancing the divisibility checker program with advanced features, user-friendly interfaces, or additional functionality to meet evolving requirements and challenges.



INetTutor.com

## Online Programming Lessons, Tutorials and Capstone Project guide

### Source code example

```
1. using System;
2.
3. namespace DivisibilityChecker
4. {
5.     class Program
6.     {
7.         static void Main(string[] args)
8.         {
9.             Console.WriteLine("iNetTutor.com");
10.            Console.WriteLine("Divisibility Checker");
11.            Console.WriteLine("-----");
12.
13.            // Ask the user for the first number
14.            Console.Write("Enter the first number: ");
15.            int number1;
16.            if (!int.TryParse(Console.ReadLine(), out number1))
17.            {
18.                Console.WriteLine("Invalid input. Please enter a valid integer.");
19.                return;
20.            }
21.
22.            // Ask the user for the second number
23.            Console.Write("Enter the second number: ");
24.            int number2;
25.            if (!int.TryParse(Console.ReadLine(), out number2) || number2 == 0)
26.            {
27.                Console.WriteLine("Invalid input. Please enter a valid non-zero
integer.");
28.                return;
29.            }
30.
31.            // Check divisibility
32.            if (number1 % number2 == 0)
33.            {
34.                Console.WriteLine($"{number1} is divisible by {number2}.");
35.            }
36.            else
37.            {
38.                Console.WriteLine($"{number1} is not divisible by {number2}.");
39.            }
40.
41.            Console.WriteLine("Press any key to exit.");
42.            Console.ReadKey();
43.        }
44.    }
45. }
```



INetTutor.com

# Online Programming Lessons, Tutorials and Capstone Project guide

## Explanation

Here's an explanation of the key sections and code segments:

### 1. Namespace and Class Definition:

- The code is wrapped in the DivisibilityChecker namespace, and it defines a class named Program.

### 2. Main Method:

- The program's execution starts in the Main method.

### 3. User Input and Validation:

- The program first displays a title and prompts the user for the dividend (the number to be checked for divisibility).
- It uses `int.TryParse` to read the user's input and validate that it's a valid integer. If not, it displays an error message and exits.

### 4. Divisor Input and Validation:

- Next, the program asks for the divisor (the number by which to check divisibility).
- It again uses `int.TryParse` to validate the input, ensuring it's a valid non-zero integer.

### 5. Divisibility Check:

- The program then performs the divisibility check using the modulo operator `%`. It calculates the remainder when dividend is divided by divisor. If the remainder is 0, it means the dividend is divisible by the divisor.

### 6. Result Display:

- Depending on the result of the divisibility check, the program displays a message indicating whether the dividend is divisible by the divisor or not.

### 7. Exiting the Program:

- Finally, the program prompts the user to press any key to exit, ensuring that the console window doesn't close immediately.




iNetTutor.com

## Online Programming Lessons, Tutorials and Capstone Project guide

This program essentially takes two integer inputs from the user, performs a divisibility check, and provides feedback on whether one number is divisible by another. It incorporates user input validation to handle invalid inputs gracefully.

### Output

 C:\Users\fujitsu\source\repos\DivisibilityChecker'

```
iNetTutor.com
Divisibility Checker
-----
Enter the first number: 100
Enter the second number: 5
100 is divisible by 5.
Press any key to exit.
```

### Summary

Recap of Key Concepts: In this lesson, we explored a simple C# console program that checks whether one number is divisible by another. Here are the key concepts covered:

- **Divisibility:** We learned the concept of divisibility in mathematics, where one number can be evenly divided by another without leaving a remainder.
- **User Input:** The program demonstrated how to obtain user input for both the dividend (the number to be checked) and the divisor (the number by which to check divisibility).
- **Input Validation:** Input validation was applied to ensure that the user's inputs are valid integers and that the divisor is a non-zero integer to avoid mathematical errors.
- **Modulo Operator (%):** We used the modulo operator % to perform the actual divisibility check. If the remainder is zero, it indicates that one number is divisible by another.
- **Conditional Statements:** Conditional statements (if and else) were used to display appropriate messages based on the divisibility check result.

Encouragement to Experiment: To further enhance your understanding of C# programming and the concept of divisibility checking, I encourage you to experiment with the code. Here are some ideas to explore:

- Try different dividend and divisor values to see how the program responds.
- Modify the program to handle additional types of numbers, such as floating-point numbers.



INetTutor.com

## Online Programming Lessons, Tutorials and Capstone Project guide

- Create a loop to allow the user to perform multiple divisibility checks without restarting the program.
- Implement more advanced divisibility rules, such as checking for divisibility by prime numbers.

By experimenting with the code and exploring variations, you can deepen your knowledge of C# programming and develop practical problem-solving skills. Don't hesitate to adapt and expand upon this program to tackle more complex divisibility-related challenges. Happy coding!

### Quiz

- 1. What is the purpose of the Divisibility Checker program in C#?**
  - A. To calculate the sum of two numbers
  - B. To determine if one number is divisible by another
  - C. To find the square root of a number
  - D. To sort an array of numbers
- 2. What does the modulo operator (%) do in the context of divisibility checking?**
  - A. It multiplies two numbers.
  - B. It adds two numbers.
  - C. It checks if one number is evenly divisible by another.
  - D. It calculates the average of two numbers.
- 3. In the program, what happens if the user enters a non-integer value for the first number?**
  - A. The program displays "Invalid input" and continues.
  - B. The program calculates the sum of the two numbers.
  - C. The program displays "Invalid input" and exits.
  - D. The program asks the user to re-enter the first number.
- 4. What does it mean if the program displays "{number1} is divisible by {number2}"?**
  - A. The program encountered an error.
  - B. number1 is not divisible by number2.
  - C. number1 is divisible by number2 without a remainder.
  - D. The program is about to exit.
- 5. What happens if the user enters a divisor (second number) with a value of zero?**
  - A. The program continues normally.
  - B. The program displays an error message and exits.
  - C. The program performs division by zero without issues.
  - D. The program asks the user to enter a valid divisor.