How to Create Simple Calculator in CSharp

Welcome to a practical guide on creating a Simple Calculator using C# programming. In this tutorial, we'll break down the process into easy-to-follow steps, making it accessible for beginners. You don't need to embark on a complex coding journey; we'll simplify everything for you. This versatile tool is not only a valuable addition to your programming skills but also a helpful companion for everyday calculations. So, let's get started and learn how to build this straightforward C# calculator from scratch.

Introduction

In this lesson, we'll create a simple C# program for a basic calculator. The program's main goal is to help users perform common math operations like addition, subtraction, multiplication, and division with ease. By the end of this lesson, you'll have the skills to build your own calculator and gain a deeper understanding of programming.

Calculators are everywhere in our daily lives, from calculating expenses to solving math problems. In programming, knowing how to create a basic calculator is a fundamental skill. It simplifies tasks and can be applied to various projects. Whether you're new to C# or an experienced developer, building a calculator is a useful skill to have. Let's dive into C# and create your calculator from scratch.

Objectives

Here are the objectives for this lesson on creating a basic calculator in C#:

1. Learn: Understand the fundamentals of C# programming, including user input, variables, and the switch statement.
2. Apply: Implement a basic calculator program that can perform addition, subtraction, multiplication, and division based on user input.
3. Create: Develop a functional C# calculator program that provides a user-friendly interface for performing arithmetic operations.
4. Enhance: Extend the calculator's functionality by adding more operations or improving its user interface.

These objectives will guide your learning journey and help you achieve a well-rounded understanding of C# programming while building a practical application.

Source code example

```
using System;

namespace SimpleCalculator
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("iNetTutor.com");
            Console.WriteLine("Simple Calculator");
            Console.WriteLine("-----------------");

            // Ask the user to enter the first number
            Console.Write("Enter the first number: ");
            double num1;
            if (!double.TryParse(Console.ReadLine(), out num1))
            {
                Console.WriteLine("Invalid input. Please enter a valid number.");
                return;
            }

            // Ask the user to enter the second number
            Console.Write("Enter the second number: ");
            double num2;
            if (!double.TryParse(Console.ReadLine(), out num2))
            {
                Console.WriteLine("Invalid input. Please enter a valid number.");
                return;
            }

            // Ask the user to select an operator
            Console.WriteLine("Select an operator:");
            Console.WriteLine("1. Addition (+)");
            Console.WriteLine("2. Subtraction (-)");
            Console.WriteLine("3. Division (/)");
            Console.WriteLine("4. Multiplication (*)");
            Console.Write("Enter the operator (1/2/3/4): ");

            char operatorSymbol = Console.ReadKey().KeyChar;
            Console.WriteLine(); // Move to the next line after user input

            double result = 0;

            // Perform the operation based on the selected operator
            switch (operatorSymbol)
            {
                case '1':
                    result = num1 + num2;
                    Console.WriteLine($"Result: {num1} + {num2} = {result}");
                    break;
                case '2':
                    result = num1 - num2;
                    Console.WriteLine($"Result: {num1} - {num2} = {result}");
                    break;
                case '3':
                    if (num2 == 0)
                    {
                        Console.WriteLine("Error: Division by zero is not allowed.");
                        return;
                    }
                    result = num1 / num2;
                    Console.WriteLine($"Result: {num1} / {num2} = {result}");
                    break;
                case '4':
                    result = num1 * num2;
                    Console.WriteLine($"Result: {num1} * {num2} = {result}");
                    break;
                default:
                    Console.WriteLine("Invalid operator selection. Please choose 1, 2, 3, or 4.");
                    break;
            }

            Console.WriteLine("Press any key to exit.");
            Console.ReadKey();
        }
    }
}
```

Explanation

This C# program is a simple calculator that allows the user to perform basic arithmetic operations (addition, subtraction, multiplication, and division) on two numbers. Here's a step-by-step explanation and discussion of the source code:

1. Namespace and Class: The program is defined within the SimpleCalculator namespace and the Program class.

2. Initialization: The program starts by displaying some introductory messages, including the program name and a brief description.

3. User Input: It then prompts the user to enter two numbers. It uses the Console.Write method to request input and Console.ReadLine() to read the input as a string. The input is converted to a double using double.TryParse(), which also performs input validation to ensure the user enters a valid number. If the input is invalid, an error message is displayed, and the program exits.

4. Operator Selection: After successfully getting the two numbers, the program asks the user to select an operator by displaying a menu of choices (addition, subtraction, division, or multiplication). The user is prompted to enter the operator as a single character (1, 2, 3, or 4) using Console.ReadKey().KeyChar. The selected operator is stored in the operatorSymbol variable.

5. Calculations: The program calculates the result based on the selected operator using a switch statement. Depending on the operator chosen, it performs the corresponding operation (addition, subtraction, division, or multiplication) on the two input numbers and stores the result in the result variable.

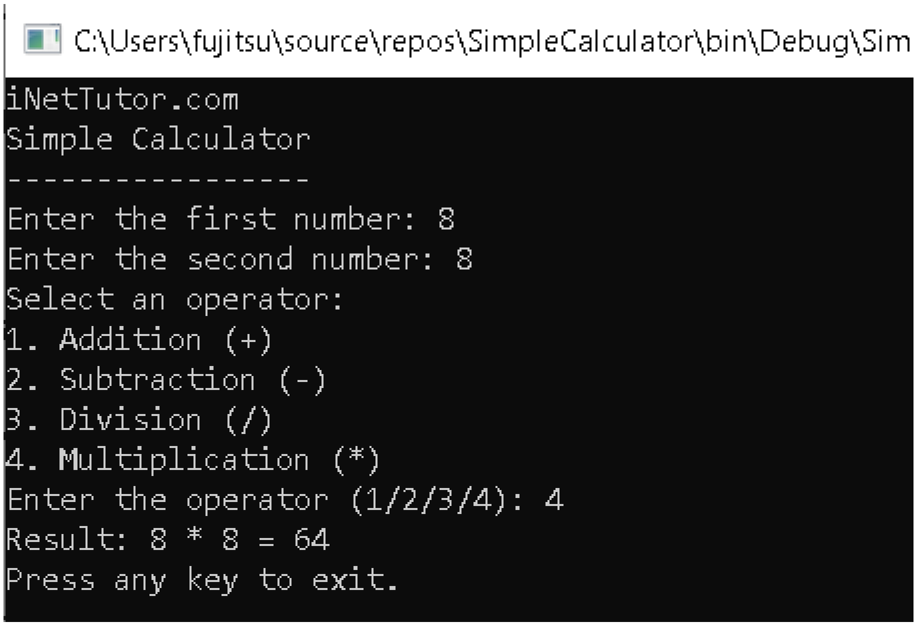
6. Output: The program then displays the result of the operation to the console, including a formatted message that shows the equation and the calculated result.

7. Error Handling: There is error handling in place, such as checking for division by zero when the user selects division. If division by zero is detected, an error message is displayed, and the program exits.

8. Exit: Finally, the program prompts the user to press any key to exit, and it waits for a key press using Console.ReadKey().

In summary, this C# program demonstrates basic console input and output, user input validation, and the use of a switch statement to perform different operations based on user input. It serves as a simple but functional calculator for performing basic arithmetic calculations.

Output



C:\Users\fujitsu\source\repos\SimpleCalculator\bin\Debug\Sim

```
iNetTutor.com
Simple Calculator
-----------------
Enter the first number: 8
Enter the second number: 8
Select an operator:
1. Addition (+)
2. Subtraction (-)
3. Division (/)
4. Multiplication (*)
Enter the operator (1/2/3/4): 4
Result: 8 * 8 = 64
Press any key to exit.
```

Summary

In this lesson, you've learned how to create a simple console-based calculator in C#. Key concepts covered include:

- User Input: You learned how to prompt the user for input and validate it to ensure it's a valid number. Input validation is crucial for preventing errors in calculations.
- Switch Statement: The lesson introduced the switch statement, which allows you to perform different actions based on the user's choice. In this program, it was used to select and execute arithmetic operations.
- Arithmetic Operations: You explored the fundamental arithmetic operations in C#—addition, subtraction, multiplication, and division. These operations are the building blocks of most calculator applications.
- Error Handling: The program included error handling to prevent issues like division by zero. This ensures that the program can handle unexpected situations gracefully.
- Output Formatting: You saw how to format and display output messages to provide clear and informative results to the user.

Encouragement to Experiment: we encourage you to experiment with this basic calculator program. Try adding more operators or expanding the menu to include additional mathematical functions. You can also enhance the program's error handling or create a graphical user interface (GUI) version. Learning by doing is a powerful way to reinforce your programming skills, so don't hesitate to explore and improve upon this foundation.

Quiz

1. What type of numbers does the program prompt the user to enter?
a) Whole numbers only
b) Decimal numbers only
c) Either whole or decimal numbers
d) Complex numbers

2. Which operator is represented by the symbol '*' in the program?
a) Addition
b) Subtraction
c) Multiplication
d) Division

3. If the user attempts to divide a number by zero, what error message does the program display?
a) "Invalid input. Please enter a valid number."
b) "Error: Division by zero is not allowed."
c) "Invalid operator selection. Please choose 1, 2, 3, or 4."
d) "Press any key to exit."

4. Which part of the program is responsible for performing the actual arithmetic operation?
a) The section asking for the first number
b) The section asking for the second number
c) The section displaying operator choices
d) The switch statement based on the selected operator

5. What does the program do if the user enters an invalid operator choice, such as '5'?
a) It performs the addition operation.
b) It displays an error message.
c) It exits the program.
d) It prompts the user to enter the operator again.