



INetTutor.com

# Online Programming Lessons, Tutorials and Capstone Project guide

## Sum of Even Numbers in CSharp

### Introduction

In the world of computer programming, the ability to calculate sums and averages is fundamental, forming the backbone of various computational tasks. Whether analyzing data, optimizing algorithms, or generating statistics, these operations are omnipresent. This lesson delves into a specific facet of this computational prowess, focusing on the nuanced process of calculating the sum and average of even numbers within a given range. Using the C# programming language, we will explore a concise yet powerful code implementation that not only reinforces the principles of iteration and mathematical computation but also equips you with practical skills applicable across a spectrum of programming scenarios. Join us on this journey as we unravel the intricacies of sum and average calculations, demonstrating their significance in the vibrant landscape of programming with C#.

### Learning Objectives

1. Explore Looping Concepts:
  - Gain a solid understanding of loops and their significance in programming.
  - Recognize how loops facilitate efficient iteration through a range of numbers.
2. Master User Input and Output Techniques in C#:
  - Learn to interact with users through the console, acquiring essential skills in user input and output operations.
  - Understand the role of the `Console.ReadLine()` method in capturing user input.
3. Validate User Input for Robust Programs:
  - Implement robust input validation techniques to ensure users provide meaningful and valid data.
  - Explore conditional statements and loops to create a responsive user input validation mechanism.
4. Calculate the Sum of Even Numbers:
  - Dive into the logic of summing even numbers within a specified range.
  - Understand the iterative process of adding even numbers and storing the cumulative sum.
5. (Optional) Calculate the Average of Even Numbers:



- Extend your understanding to calculate the average of even numbers within the given range.
- Explore the relationship between the sum and the count of even numbers to determine the average.

By the end of this lesson, you will not only grasp the core concepts of loops, user input/output, and input validation but also possess the practical skills to develop a C# program that calculates the sum and average of even numbers. These skills lay a strong foundation for tackling diverse programming challenges and form an integral part of your programming toolkit.

### Source code example

```
1. using System;
2.
3. namespace NumberSumCalculator
4. {
5.     class Program
6.     {
7.         static void Main(string[] args)
8.         {
9.             Console.WriteLine("iNetTutor.com");
10.            // Initialize sum and average variables
11.            int sum = 0;
12.            double average = 0.0;
13.
14.            // User prompt and input validation
15.            bool validInput = false;
16.            int limit = 0;
17.            while (!validInput)
18.            {
19.                Console.WriteLine("Enter the upper limit for even numbers: ");
20.                string input = Console.ReadLine();
21.                if (int.TryParse(input, out limit) && limit > 1)
22.                {
23.                    validInput = true;
24.                }
25.                else
26.                {
27.                    Console.WriteLine("Invalid input. Please enter a whole number
greater than 1.");
28.                }
29.            }
30.
31.            // Loop through and calculate sum and count
32.            int count = 0;
33.            for (int i = 2; i < limit; i += 2)
34.            {
35.                sum += i;
36.                count++;
37.            }
38.
39.            // Calculate and display average
```



```
40.         if (count > 0)
41.         {
42.             average = (double)sum / count;
43.             Console.WriteLine($"The sum of even numbers from 2 to {limit} is:
         {sum}");
44.             Console.WriteLine($"The average of even numbers from 2 to {limit} is:
         {average:F2}");
45.             Console.ReadKey();
46.         }
47.         else
48.         {
49.             Console.WriteLine("No even numbers within the specified range.");
50.             Console.ReadKey();
51.         }
52.     }
53. }
54. }
```

### Explanation

Namespace, Class Declaration, and Variable Initialization (line 1-16):

- This program is encapsulated within the NumberSumCalculator namespace, with the main logic residing in the Program class. It starts by initializing variables for the sum (int) and average (double) of even numbers.

User Input and Validation (line 17-29):

- A while loop handles user input and validation. The user is prompted to enter the upper limit for even numbers, and the input is checked to ensure it's a whole number greater than 1. The loop continues until valid input is received.

Loop for Sum and Count (line 32-37):

- A for loop iterates through even numbers from 2 to the specified limit (excluding the limit itself). Within the loop, each even number is added to the sum, and the count is incremented.

Calculate and Display Average (line 40-51):

- After the loop, the program calculates the average if there are any even numbers within the specified range. It then displays the sum and average with appropriate formatting. If no even numbers are found, a message is displayed.

This code serves as a practical example of user input validation, iterative calculation, and conditional logic in C#. It enables the user to determine the sum and average of even numbers within a given range, contributing to a foundational understanding of control flow and mathematical computation in programming.



iNetTutor.com

## Online Programming Lessons, Tutorials and Capstone Project guide

### Output

```
C:\Users\fujitsu\source\repos\SumEvenNumbers\bin\Debug\SumEvenNumbers.exe
iNetTutor.com
Enter the upper limit for even numbers:
10
The sum of even numbers from 2 to 10 is: 20
The average of even numbers from 2 to 10 is: 5.00
```

### Summary

In this C# console programming lesson, we delved into the fundamental concepts of user input validation, iterative calculation, and conditional logic through the lens of calculating the sum and average of even numbers within a specified range. Beginning with the initialization of sum and average variables, the program seamlessly navigated through the intricacies of user interaction, employing a while loop for input validation to ensure the user provides meaningful and valid data.

The subsequent for loop efficiently computed the sum and count of even numbers within the user-defined range, demonstrating the elegance of iterative calculations. The program also showcased the optional calculation of the average, expanding the understanding of mathematical operations in programming.

Moreover, this lesson underscored the significance of user-friendly interfaces, employing concise error messages to guide users towards valid inputs. The interplay between user input, loops, and mathematical computations illuminated core programming principles essential for building robust and interactive applications.

By mastering these concepts, learners not only gained proficiency in developing a program that solves a specific problem but also acquired transferable skills applicable to a broader spectrum of programming challenges. This lesson serves as a stepping stone for individuals aspiring to build a strong foundation in C# programming, providing valuable insights into the realms of control flow, mathematical operations, and user-centric design.

### Exercises

#### 1. Odd Numbers Calculation:

- Modify the program to calculate the sum and average of odd numbers within the specified range. Adjust the loop accordingly.



INetTutor.com

## Online Programming Lessons, Tutorials and Capstone Project guide

### 2. Enhanced User Interaction:

- Improve the user interface by adding instructions, clarifying messages, or providing additional information during the input and output phases.

### 3. Dynamic Upper Limit:

- Allow the user to decide whether to include or exclude the upper limit in the calculation. Implement the necessary adjustments in the loop logic.

### 4. Input Validation Enhancement:

- Implement more robust input validation. Ensure that the program handles a broader range of user inputs gracefully, providing clear feedback in case of errors.

### 5. Multiple Range Calculation:

- Extend the program to allow users to calculate the sum and average for multiple ranges in a single run. Provide a mechanism to input and process multiple range calculations.

### Assessment:

#### 1. Code Review:

- Conduct a code review session, focusing on the clarity, readability, and structure of the code. Discuss potential improvements and optimizations.

#### 2. Error Handling Assessment:

- Purposefully introduce errors in user input and evaluate how well the program handles and communicates these errors to the user. Provide feedback on enhancing error-handling mechanisms.

#### 3. Efficiency Analysis:

- Evaluate the efficiency of the program, especially regarding the calculation of the sum and average. Discuss potential optimizations and improvements in terms of algorithmic efficiency.

#### 4. User Experience Evaluation:

- Assess the overall user experience by gathering feedback on the program's interface, instructions, and clarity of output. Use this feedback to enhance user interaction.

#### 5. Scenario-Based Assessment:



INetTutor.com

## Online Programming Lessons, Tutorials and Capstone Project guide

- Present learners with different scenarios and ask them to adapt the program to meet specific requirements. This could include varying the data types, adjusting the range, or introducing additional calculations.

These exercises and assessments aim to challenge learners to extend and refine the existing project, reinforcing their understanding of key concepts while encouraging creativity and problem-solving skills.

### **Meta Description**

"Master C# programming: Calculate sum and average of even numbers. Enhance skills with exercises and assessments. Explore user-friendly coding concepts."