



INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

BMI Calculator in C#

Introduction

Welcome to the next topic in our C# tutorial series! In this lesson, we will explore how to create a Body Mass Index (BMI) Calculator using C#. The BMI Calculator is a useful tool that calculates a person's BMI based on their height and weight. It then categorizes them as underweight, normal weight, overweight, or obese based on the BMI range.

BMI is a widely used measure to assess whether a person has a healthy body weight for their height. It provides an indication of the relative healthiness of an individual and is used by healthcare professionals to discuss body weight objectively with their patients. By understanding how to calculate and categorize BMI, you will gain valuable insights into the relationship between weight and health.

Throughout this lesson, we will guide you through the process of creating a BMI Calculator program step by step. We will explain the underlying concepts, provide code examples, and encourage you to practice and apply your knowledge to reinforce your understanding.

Objectives

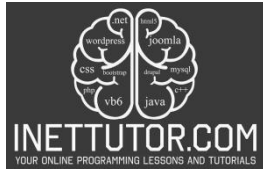
In this lesson, our primary focus is to help you understand, learn, practice, and apply the concepts of BMI calculation and categorization using C#. By the end of this lesson, you will be able to:

1. Understand the concept of BMI and its significance in assessing body weight.
2. Learn how to design and structure a BMI Calculator program in C#.
3. Practice implementing the BMI calculation formula using appropriate mathematical operations in C#.
4. Apply if-else conditions to categorize individuals based on their BMI range and display the appropriate message for each category.

By achieving these objectives, you will gain a deeper understanding of BMI, enhance your programming skills in C#, and be able to create a BMI Calculator that can provide valuable insights into an individual's weight status.

Source code example

```
1. using System;
2.
3. namespace BMICalculator
4. {
5.     class Program
6.     {
7.         static void Main()
```

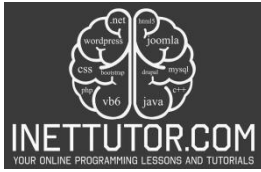


INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

```
8.      {
9.          Console.WriteLine("iNetTutor.com - BMI Calculator");
10.
11.         Console.Write("Enter your height in meters: ");
12.         double height = double.Parse(Console.ReadLine());
13.
14.         Console.Write("Enter your weight in kilograms: ");
15.         double weight = double.Parse(Console.ReadLine());
16.
17.         double bmi = weight / (height * height);
18.
19.         string weightStatus;
20.
21.         if (bmi < 18.5)
22.         {
23.             weightStatus = "Underweight";
24.         }
25.         else if (bmi >= 18.5 && bmi < 25)
26.         {
27.             weightStatus = "Normal weight";
28.         }
29.         else if (bmi >= 25 && bmi < 30)
30.         {
31.             weightStatus = "Overweight";
32.         }
33.         else
34.         {
35.             weightStatus = "Obese";
36.         }
37.
38.         Console.WriteLine($"Your BMI: {bmi:F2}");
39.         Console.WriteLine($"Weight Status: {weightStatus}");
40.
41.         Console.ReadLine();
42.     }
43. }
44. }
```

```
01. using System;
02.
03. namespace BMI Calculator
04. {
05.     class Program
06.     {
07.         static void Main()
08.         {
09.             Console.WriteLine("iNetTutor.com - BMI Calculator");
10.
11.             Console.Write("Enter your height in meters: ");
12.             double height = double.Parse(Console.ReadLine());
13.
14.             Console.Write("Enter your weight in kilograms: ");
15.             double weight = double.Parse(Console.ReadLine());
16.
17.             double bmi = weight / (height * height);
18.
19.             string weightStatus;
20.
21.             if (bmi < 18.5)
22.             {
23.                 weightStatus = "Underweight";
24.             }
25.             else if (bmi >= 18.5 && bmi < 25)
26.             {
27.                 weightStatus = "Normal weight";
28.             }
29.             else if (bmi >= 25 && bmi < 30)
30.             {
31.                 weightStatus = "Overweight";
32.             }
33.             else
34.             {
35.                 weightStatus = "Obese";
36.             }
37.
38.             Console.WriteLine($"Your BMI: {bmi:F2}");
39.             Console.WriteLine($"Weight Status: {weightStatus}");
40.
41.             Console.ReadLine();
42.         }
43.     }
44. }
```



INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

Explanation

The code you provided calculates the Body Mass Index (BMI) of a user and categorizes their weight based on the calculated BMI. Here's a breakdown of the code:


1. The using System; statement is used to include the System namespace, which contains the fundamental classes and base types that are commonly used in C# programs.
2. The namespace BMI Calculator statement defines a namespace called "BMI Calculator". Namespaces are used to organize and group related code.
3. Inside the BMI Calculator namespace, we have a class Program declaration. This class contains the main entry point of the program.
4. The static void Main() method is the entry point of the program. It is the method that gets executed when the program starts running.
5. Console.WriteLine("iNetTutor.com - BMI Calculator"); is used to display a message to the console, welcoming the user to the BMI Calculator program.
6. Console.Write("Enter your height in meters: "); prompts the user to enter their height in meters by displaying a message without a new line.
7. double height = double.Parse(Console.ReadLine()); reads the user's input from the console and stores it in the height variable after converting it to a double data type.
8. Console.Write("Enter your weight in kilograms: "); prompts the user to enter their weight in kilograms.
9. double weight = double.Parse(Console.ReadLine()); reads the user's input for weight and stores it in the weight variable after converting it to a double data type.
10. double bmi = weight / (height * height); calculates the BMI by dividing the weight by the square of the height.
11. string weightStatus; declares a variable called weightStatus of type string to store the weight status category.
12. The if-else conditions are used to categorize the BMI and assign the appropriate weight status to the weightStatus variable. If the BMI is less than 18.5, it is considered underweight. If the BMI is between 18.5 (inclusive) and 25 (exclusive), it is considered normal weight. If the BMI is between 25 (inclusive) and 30 (exclusive), it is considered overweight. Any BMI above 30 is classified as obese.
13. Console.WriteLine(\$"Your BMI: {bmi:F2}"); displays the calculated BMI using string interpolation. The :F2 format specifier is used to display the value with two decimal places.
14. Console.WriteLine(\$"Weight Status: {weightStatus}"); displays the weight status message using string interpolation and the weightStatus variable.
15. Console.ReadLine(); waits for the user to press Enter before the program exits.

Output



iNetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

 C:\Users\fujitsu\source\repos\BMICalculator_CSharp\bin\Debug\BMICalculator_CSharp.exe

```
iNetTutor.com - BMI Calculator
Enter your height in meters: 1.5
Enter your weight in kilograms: 60
Your BMI: 26.67
Weight Status: Overweight
```

Summary

In this BMI Calculator lesson, we crafted a simple C# program that calculates Body Mass Index (BMI) based on user-provided height and weight. The program then categorizes the BMI into different weight statuses: Underweight, Normal weight, Overweight, or Obese. The user-friendly console application not only computes and displays the BMI but also communicates the individual's weight status effectively.

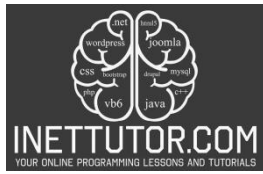
This practical lesson enhances the understanding of basic arithmetic operations and conditional statements in C#. By dissecting the source code, learners grasp fundamental concepts such as user input handling, mathematical computations, and decision-making structures. The BMI Calculator serves as a stepping stone for beginners to reinforce their programming skills and gain hands-on experience with real-world applications. Aspiring developers can further enrich their knowledge by exploring additional features or refining the program's user interface. Overall, this lesson provides a foundational understanding of BMI calculations and encourages learners to apply programming concepts to solve everyday problems.

Exercises and Assessment

The provided C# program for a BMI calculator is a great starting point. Here are some activities and exercises to enhance its functionality and user experience:

- Error handling: Implement checks for invalid input (e.g., non-numeric values, negative values) and provide informative error messages.
- Output formatting: Refine the BMI and weight status output. Use string formatting or tables to present the information more clearly and visually appealing.
- Category customization: Allow users to choose between different BMI classification standards by providing options and dynamically adjusting category ranges.
- Unit conversion: Add features to convert height and weight between different units (e.g., meters to feet, kilograms to pounds) to cater to wider user preferences.

Meta Description



INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

Learn C# by building a BMI Calculator. Understand BMI, apply if-else conditions, and categorize weight status. Practical coding for health awareness.