

INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

Circle Properties Calculator in CSharp

Introduction

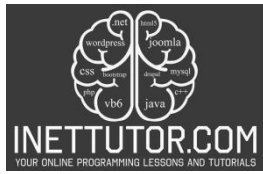
Welcome to the world of C# programming! In this lesson, we're going to create a tool called the Circle Properties Calculator. But what does it do? Well, think of it as a helpful tool for finding out things about circles. You know, those round shapes? Specifically, it'll figure out the diameter, area, and how long around the edge (circumference) a circle is. You tell it how big the circle is by giving it a number called the radius, and it'll do all the math for you.

You might wonder why it's useful to know stuff about circles. The truth is, circles are all around us, not just in textbooks. They show up in lots of real-life situations. For example, architects use circles when they design round parts of buildings or roads. Engineers need to get circle math right for things like making gears in machines. Even in everyday life, like figuring out how much pizza you have or how far you run around a circular track, understanding circle properties comes in handy. So, by the end of this lesson, you'll not only have a nifty Circle Properties Calculator but also see how these circle calculations can help you in everyday situations. Let's get started!

Objectives

Here are four outcomes-based objectives for the "Circle Properties Calculator in C#" lesson:

1. **Learn:** By the end of this lesson, learners will have acquired a solid understanding of key mathematical concepts related to circles, including radius, diameter, area, and circumference.
2. **Apply:** Students will be able to apply their knowledge to solve real-world problems that involve circular shapes, such as calculating the dimensions of circular objects or spaces.
3. **Create:** Learners will demonstrate their programming skills by creating a functional Circle Properties Calculator program in C#, which can take user input and perform calculations to determine circle properties accurately.
4. **Upgrade:** With a strong foundation in circle properties, students will have the skills and knowledge to enhance and expand the Circle Properties Calculator, adding additional features or improving its user interface for more advanced applications.



Source code example

```
1. using System;
2.
3. namespace CirclePropertiesCalculator
4. {
5.     class Program
6.     {
7.         static void Main(string[] args)
8.         {
9.             Console.WriteLine("iNetTutor.com");
10.            Console.WriteLine("Circle Properties Calculator");
11.
12.            // Ask the user to enter the radius
13.            Console.Write("Enter the radius of the circle: ");
14.
15.            // Read user input and convert it to a double
16.            if (double.TryParse(Console.ReadLine(), out double radius) && radius >= 0)
17.            {
18.                // Calculate diameter, area, and circumference
19.                double diameter = 2 * radius;
20.                double area = Math.PI * Math.Pow(radius, 2);
21.                double circumference = 2 * Math.PI * radius;
22.
23.                // Display the results
24.                Console.WriteLine($"Diameter: {diameter}");
25.                Console.WriteLine($"Area: {area}");
26.                Console.WriteLine($"Circumference: {circumference}");
27.            }
28.            else
29.            {
30.                Console.WriteLine("Invalid input. Please enter a valid non-negative
number for the radius.");
31.            }
32.
33.            Console.WriteLine("Press any key to exit.");
34.            Console.ReadKey();
35.        }
36.    }
37. }
```

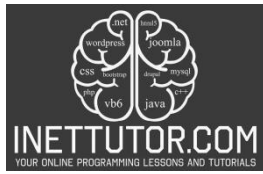
Explanation

This C# program is designed to calculate and display the properties of a circle based on user input, specifically the radius of the circle. Let's break down the source code step by step:

1. Using Statements:

- `using System;`: This statement includes the System namespace, which provides access to fundamental classes and functionality.

2. Namespace:



INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

- namespace CirclePropertiesCalculator: The program is organized within a namespace called CirclePropertiesCalculator, which helps in organizing related classes and program elements.

3. Main Method:

- static void Main(string[] args): This is the entry point of the program, where execution begins when the program is run. It accepts an array of string arguments args, which are typically used for command-line input.

4. Display Information:

- Console.WriteLine("iNetTutor.com"): This line displays the program's name or identifier, which is "iNetTutor.com."
- Console.WriteLine("Circle Properties Calculator"): This line displays a title indicating the purpose of the program, which is to calculate circle properties.

5. User Input for Radius:

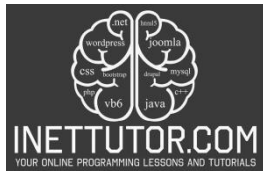
- Console.Write("Enter the radius of the circle: "): This line prompts the user to enter the radius of the circle by displaying a message.
- if (double.TryParse(Console.ReadLine(), out double radius) && radius >= 0): It reads the user's input as a string using Console.ReadLine(), tries to parse it as a double, and stores the result in the radius variable. The && radius >= 0 condition checks if the entered value is non-negative.

6. Calculations:

- Inside the if block, calculations are performed based on the entered radius:
 - double diameter = 2 * radius;: Calculates the diameter of the circle.
 - double area = Math.PI * Math.Pow(radius, 2);: Computes the area of the circle using the formula πr^2 .
 - double circumference = 2 * Math.PI * radius;: Computes the circumference of the circle using the formula $2\pi r$.

7. Display Results:

- Console.WriteLine(\$"Diameter: {diameter}"): Displays the calculated diameter.
- Console.WriteLine(\$"Area: {area}"): Displays the calculated area.



iNetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

- `Console.WriteLine($"Circumference: {circumference}")`: Displays the calculated circumference.

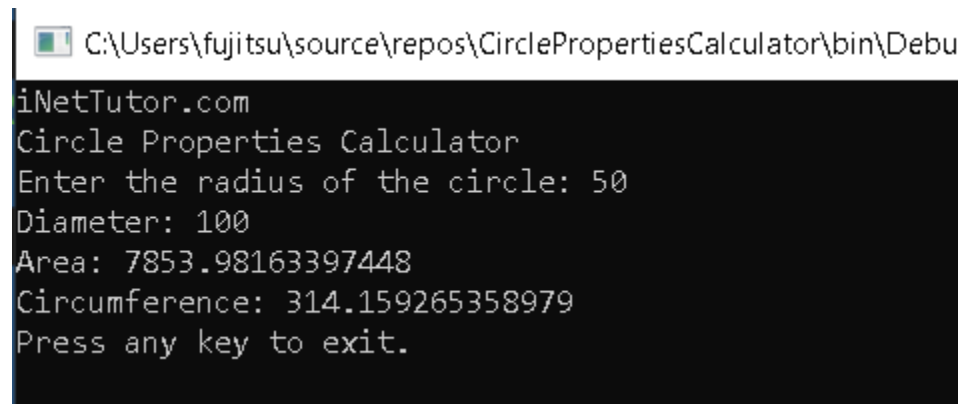
8. Invalid Input Handling:

- If the user enters invalid input (e.g., non-numeric or negative radius), the program will display the message: "Invalid input. Please enter a valid non-negative number for the radius."

9. Program Termination:

- `Console.WriteLine("Press any key to exit.")`: Informs the user to press any key to exit the program.
- `Console.ReadKey();`: Waits for the user to press a key before the program exits.

Output



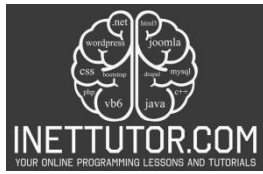
```
C:\Users\fujitsu\source\repos\CirclePropertiesCalculator\bin\Debu
iNetTutor.com
Circle Properties Calculator
Enter the radius of the circle: 50
Diameter: 100
Area: 7853.98163397448
Circumference: 314.159265358979
Press any key to exit.
```

Summary

In this blog post, we embarked on a journey to demystify the process of creating a Circle Properties Calculator in C#. We learned how to create a simple yet powerful console application that calculates and displays the diameter, area, and circumference of a circle based on user input.

Key Concepts Recap:

- We explored the fundamental structure of a C# console application, including namespaces, the Main method, and the usage of the Console class for input and output operations.
- The program prompted the user to enter the radius of the circle, showcasing the interaction between the program and the user.



INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

- We demonstrated the importance of input validation to ensure that the entered radius is a valid non-negative number.
- Calculations for the diameter, area, and circumference were carried out using basic mathematical formulas, demonstrating how C# can perform mathematical operations.
- The results were displayed in a clear and user-friendly format, enhancing the program's readability and usefulness.

Encouragement to Experiment: I encourage you to take this code as a starting point for your C# programming journey. Experiment with it, modify it, and explore variations to gain a deeper understanding of C# and programming in general. Try different input validation techniques, formatting options, or even extend the program to calculate additional circle properties. Learning by doing is the best way to enhance your programming skills, so don't hesitate to embark on your coding adventure with this Circle Properties Calculator!

In the world of programming, there are countless possibilities, and your creativity knows no bounds. Whether you're a beginner or an experienced coder, the journey of exploration and experimentation is a valuable one. Happy coding!

Exercises and Assessment

Exercises and Assessment for Improving the Circle Properties Calculator:

Exercise 1: Validate Radius

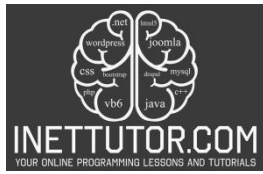
- Modify the program to provide more detailed error messages when invalid input is entered.
- Assess: Check if the program now provides specific error messages for different types of invalid input (negative numbers, non-numeric characters, etc.).

Exercise 2: User Interaction

- Enhance the user interaction by allowing users to input multiple circle radii and calculating properties for each one.
- Assess: Verify if the program can handle multiple radius calculations and display the results correctly.

Exercise 3: Additional Properties

- Extend the program to calculate additional circle properties, such as the area of a sector or the length of an arc based on user input.



- Assess: Ensure that the program correctly calculates and displays the additional circle properties.

Exercise 4: Error Handling

- Implement more robust error handling mechanisms, including catching exceptions and providing guidance to users on how to correct their input.
- Assess: Test the program with various invalid inputs and confirm that it gracefully handles exceptions and provides helpful guidance.

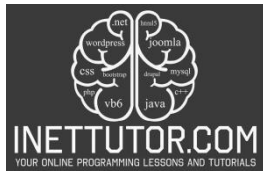
Assessment:

1. Provide a set of test cases for the improved program, including valid and invalid inputs, and evaluate if the program produces correct results and handles errors gracefully.
2. Assess the program's overall code quality, readability, and adherence to best practices.
3. Encourage learners to explain the improvements they made and their reasoning behind them.

Quiz

1. What is the main purpose of the "Circle Properties Calculator" program?
 - a) Calculate the volume of a sphere
 - b) Compute the properties of a circle based on user input
 - c) Convert units of length
 - d) Perform complex mathematical calculations
2. Which circle properties are calculated by the program?
 - a) Area and perimeter
 - b) Diameter, area, and circumference
 - c) Radius and area
 - d) Circumference and radius
3. In the code, what does Math.PI represent?
 - a) The radius of the circle
 - b) The mathematical constant π (pi)
 - c) The diameter of the circle
 - d) The circumference of the circle
4. How is the user's input for the radius validated in the program?
 - a) Using regular expressions
 - b) By checking if it's an integer
 - c) By ensuring it's a positive number
 - d) By comparing it to a predefined value

- What does the program do if the user enters an invalid input for the radius?
 - Terminates the program
 - Displays an error message and continues
 - Ignores the input and proceeds with calculations
 - Prompts the user to re-enter the radius
- How is the diameter of the circle calculated in the program?
 - By multiplying the radius by 2
 - By dividing the circumference by 2
 - By taking the square root of the radius
 - By squaring the radius
- Which C# data type is used to store the radius of the circle?
 - string
 - double
 - int
 - char
- What message does the program display before asking the user to enter the radius?
 - "Please enter the diameter of the circle:"
 - "Enter your name:"
 - "Enter the radius of the circle:"
 - "Choose an operation (add/subtract/multiply/divide):"
- What will be displayed as the program output if the user enters a radius of 5.0?
 - Diameter: 10.0 Area: 78.54 Circumference: 31.42
 - Diameter: 5.0 Area: 25.0 Circumference: 15.7
 - Diameter: 2.5 Area: 19.63 Circumference: 15.7
 - Diameter: 5.0 Area: 78.54 Circumference: 78.54
- What action does the program take after displaying the results?
 - It prompts the user to choose another operation.
 - It waits for the user to press any key before exiting.
 - It clears the console screen.
 - It automatically terminates without waiting for user input.



INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

```
1. using System;
2.
3. namespace CirclePropertiesCalculator
4. {
5.     class Program
6.     {
7.         static void Main(string[] args)
8.         {
9.             Console.WriteLine("iNetTutor.com");
10.            Console.WriteLine("Circle Properties Calculator");
11.
12.            // Ask the user to enter the radius
13.            Console.Write("Enter the radius of the circle: ");
14.
15.            // Read user input and convert it to a double
16.            if (double.TryParse(Console.ReadLine(), out double radius) && radius >= 0)
17.            {
18.                // Calculate diameter, area, and circumference
19.                double diameter = 2 * radius;
20.                double area = Math.PI * Math.Pow(radius, 2);
21.                double circumference = 2 * Math.PI * radius;
22.
23.                // Display the results
24.                Console.WriteLine($"Diameter: {diameter}");
25.                Console.WriteLine($"Area: {area}");
26.                Console.WriteLine($"Circumference: {circumference}");
27.            }
28.            else
29.            {
30.                Console.WriteLine("Invalid input. Please enter a valid non-negative number for the radius.");
31.            }
32.
33.            Console.WriteLine("Press any key to exit.");
34.            Console.ReadKey();
35.        }
36.    }
37. }
```