



INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

Age Category Checker in C#

Introduction

Welcome to the next lesson on C#! In this lesson, we will be learning about the Age Category Checker in C#. This program will allow us to determine the age category of a person based on their age.

Before we dive into the details of the Age Category Checker, let's briefly discuss C# and its importance in the world of programming.

C# is a general-purpose programming language that was developed by Microsoft in the early 2000s. It is widely used for developing a variety of applications, including desktop, web, and mobile applications. C# is known for its simplicity, readability, and object-oriented programming capabilities.

Now that we have a basic understanding of C#, let's move on to the Age Category Checker program. This program will take a person's age as input and determine which age category they fall into. The age categories can be defined based on specific criteria, such as child, teenager, adult, or senior citizen.

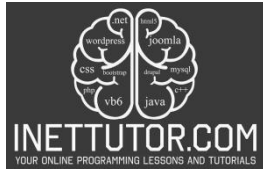
Throughout this lesson, we will explore the concepts of conditional statements, variables, and user input in C#. By the end of this lesson, you will have a solid understanding of how to create a program that categorizes individuals based on their age.

Objectives

In this lesson on the Age Category Checker in C#, we will focus on the following objectives:

1. **Understand** the concept of conditional statements in C# and their role in decision-making within a program.
2. **Learn** how to declare and initialize variables in C# to store and manipulate data.
3. **Practice** writing conditional statements using if-else statements and switch-case statements in C# to categorize individuals based on their age.
4. **Apply** the knowledge gained to create a complete Age Category Checker program that takes user input, determines the age category, and displays the result.

By the end of this lesson, you will have a solid understanding of conditional statements, variables, and user input in C#. You will be able to write a program that categorizes individuals based on their age and apply this knowledge to solve similar problems in your future projects.

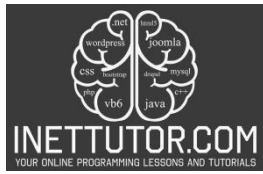


INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

Source code example

```
1. using System;
2.
3. namespace AgeCategoryChecker
4. {
5.     class Program
6.     {
7.         static void Main()
8.         {
9.             Console.WriteLine("iNetTutor.com");
10.            Console.WriteLine("Welcome to the Age Category Checker!");
11.
12.            Console.Write("Enter your age: ");
13.            int age = int.Parse(Console.ReadLine());
14.
15.            if (age >= 0 && age <= 12)
16.            {
17.                Console.WriteLine("You are a child! Enjoy your youth!");
18.            }
19.            else if (age >= 13 && age <= 19)
20.            {
21.                Console.WriteLine("Welcome to the exciting world of being a teenager!");
22.            }
23.            else if (age >= 20 && age <= 64)
24.            {
25.                Console.WriteLine("Congratulations! You are an adult now.");
26.            }
27.            else if (age >= 65)
28.            {
29.                Console.WriteLine("Welcome to the wonderful stage of being a senior citizen!");
30.            }
31.            else
32.            {
33.                Console.WriteLine("Invalid age entered. Please enter a valid age.");
34.            }
35.
36.            Console.ReadLine();
37.        }
38.    }
39. }
```



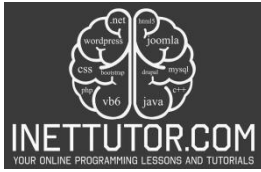
```
01. using System;
02.
03. namespace AgeCategoryChecker
04. {
05.     class Program
06.     {
07.         static void Main()
08.         {
09.             Console.WriteLine("iNetTutor.com");
10.             Console.WriteLine("Welcome to the Age Category Checker!");
11.
12.             Console.Write("Enter your age: ");
13.             int age = int.Parse(Console.ReadLine());
14.
15.             if (age >= 0 && age <= 12)
16.             {
17.                 Console.WriteLine("You are a child! Enjoy your youth!");
18.             }
19.             else if (age >= 13 && age <= 19)
20.             {
21.                 Console.WriteLine("Welcome to the exciting world of being a teenager!");
22.             }
23.             else if (age >= 20 && age <= 64)
24.             {
25.                 Console.WriteLine("Congratulations! You are an adult now.");
26.             }
27.             else if (age >= 65)
28.             {
29.                 Console.WriteLine("Welcome to the wonderful stage of being a senior citizen!");
30.             }
31.             else
32.             {
33.                 Console.WriteLine("Invalid age entered. Please enter a valid age.");
34.             }
35.
36.             Console.ReadLine();
37.         }
38.     }
39. }
```

Explanation

This C# code implements a program that categorizes users based on their age, displaying personalized messages for each category.

Breakdown:

1. Namespace and Class:
 - o using System:: Includes the System namespace, providing essential functionalities like input/output.
 - o namespace AgeCategoryChecker: Organizes the code within a specific namespace.
 - o class Program: Defines the main program class.
2. Main Method:
 - o static void Main(): The entry point of the program.
3. Welcome Message:
 - o Console.WriteLine("iNetTutor.com");: Displays a personalized message, but you can remove it if desired.



iNetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

- `Console.WriteLine("Welcome to the Age Category Checker!");`: Prints the program's purpose.
- 4. User Input:
 - `Console.Write("Enter your age: ");`: Prompts the user to enter their age.
 - `int age = int.Parse(Console.ReadLine());`: Reads the user's input, parses it into an integer, and stores it in the age variable.
- 5. Age Category Check:
 - if-else if chain: A series of conditional statements used to evaluate the user's age and assign them to a category:
 - `age >= 0 && age <= 12`: Checks if the age is between 0 and 12 (inclusive), categorizing them as a child.
 - Similar conditions exist for teenagers (13-19), adults (20-64), and senior citizens (65+).
 - Each if or else if block prints a corresponding message if the condition is met.
 - else: If the age doesn't fit any category, an error message is displayed.
- 6. Closing:
 - `Console.ReadLine();`: Pauses the program before closing, allowing the user to see the output.

Additional Notes:

- This code only uses basic if-else statements. You could explore using a switch statement for a more concise version depending on your preference.
- You can modify the age ranges and category messages to suit specific needs.
- Remember to handle potential errors, such as invalid input types or ages outside the expected range.

By understanding this code and its explanations, you gain valuable insights into C# programming fundamentals like:

- User input and output using Console class.
- Conditional statements (if-else if) for decision-making.
- Variable declaration and usage.
- Basic program structure and execution.

Output

```
C:\Users\fujitsu\source\repos\AgeCategoryChecker\bin\Debug\AgeCategoryChecker.exe
```

```
iNetTutor.com
Welcome to the Age Category Checker!
Enter your age: 35
Congratulations! You are an adult now.
```



INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

Summary

In this lesson, we covered the Age Category Checker in C#. We learned how to prompt the user for their age and categorize them into different age groups: child, teenager, adult, or senior citizen. By using if-else statements, we were able to check the age entered by the user and display the appropriate message for each category.

By completing the lesson, you should now have a solid understanding of how to use if-else statements in C# to categorize individuals based on their age. You have also gained experience in handling user input and displaying appropriate messages based on the entered age.

Remember to practice what you have learned by experimenting with different inputs and modifying the program to suit your needs. Building upon this foundation, you can continue exploring more complex programming concepts and expand your skills in C#.

Exercises and Assessment

Remember: Utilize the provided code as a starting point, but don't be afraid to rewrite or restructure it as needed. Experiment with different approaches, test your code thoroughly, and seek help if you get stuck. By tackling these exercises and the lab exam, you'll not only refine your age category checker skills but also gain valuable experience in user input, date manipulation, conditional statements, and building practical applications in C#.

These exercises and lab exam elements encourage students to not only grasp the existing concepts but also apply critical thinking to improve and expand the functionality of the Age Category Checker program.

Tasks:

- **Dynamic age ranges:** Instead of fixed ranges, allow users to input the minimum and maximum ages for each category, making the program more adaptable.
- **Multiple categories:** Add more age categories like "young adult" or "middle-aged" to provide a more nuanced classification.
- **Error handling:** Implement robust error handling to gracefully handle invalid input (e.g., non-numeric input, ages outside expected range).

Meta Description

Master C# conditional statements with the Age Category Checker. Learn to categorize users into child, teenager, adult, or senior based on their age. Real-world programming skills.