

## Reverse Pyramid in C#

### Introduction

In this lesson, we will explore the concept of creating a reverse pyramid pattern using nested for loops in C#. Loops are powerful control structures that allow us to repeat a block of code multiple times. By using nested loops, we can create complex patterns and structures.

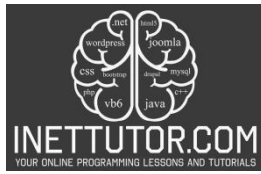
The purpose of this lesson is to demonstrate how loops and nested loops can be used to generate a reverse pyramid pattern of stars. A reverse pyramid pattern is a pattern that starts with a wide base and narrows down towards the top. It is a visually appealing pattern that can be used for decorative purposes or as a learning exercise to understand loop structures.

By understanding and implementing the reverse pyramid pattern using nested for loops, you will gain a deeper understanding of loop control structures and their ability to create intricate patterns. This lesson will enhance your logical thinking and problem-solving skills while providing hands-on experience with loops in C#.

### Objectives

The objectives of this lesson on creating a reverse pyramid pattern in C# using nested for loops are designed to provide a comprehensive learning experience. Through this lesson, you will understand the concept of loops and nested loops, learn how to implement them to create intricate patterns, practice writing code using nested loops, and apply your knowledge to generate a reverse pyramid pattern. By accomplishing these objectives, you will gain a solid foundation in loop structures, enhance your problem-solving skills, and develop the ability to create visually appealing patterns in C#.

1. Understand the concept of loops: Gain a clear understanding of how loops work and their role in repeating a block of code multiple times. Explore the different types of loops available in C# and their specific use cases.
2. Learn how to implement nested loops: Learn the syntax and logic behind nesting loops within one another. Understand how to control the flow of execution and use nested loops to create complex patterns.
3. Practice writing code with nested loops: Gain hands-on experience by practicing writing code that utilizes nested loops to generate patterns. Develop your skills in manipulating loop variables, controlling loop iterations, and managing loop termination conditions.
4. Apply knowledge to create a reverse pyramid pattern: Apply the concepts learned to create a reverse pyramid pattern using nested for loops in C#. Understand the logic and steps involved in generating the pattern, and experiment with different variations to enhance your understanding.



INetTutor.com

## Online Programming Lessons, Tutorials and Capstone Project guide

By achieving these objectives, you will have a strong grasp of loops and nested loops in C#, along with the confidence to create unique and visually appealing patterns. This lesson will equip you with valuable problem-solving and coding skills that can be applied to various programming scenarios.

### Source code example

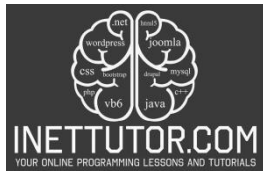
```
1. using System;
2.
3. namespace ReversePyramidPattern
4. {
5.     class Program
6.     {
7.         static void Main(string[] args)
8.         {
9.             Console.WriteLine("iNetTutor.com - Reverse Pyramid");
10.            int height = 5;
11.
12.            // Outer loop iterates for each row
13.            for (int i = height; i >= 1; i--)
14.            {
15.                // Inner loop prints spaces until reaching the current row number
16.                for (int j = 1; j <= height - i; j++)
17.                {
18.                    Console.Write(" ");
19.                }
20.
21.                // Inner loop prints stars for the current row
22.                for (int k = 1; k <= 2 * i - 1; k++)
23.                {
24.                    Console.Write("*");
25.                }
26.
27.                Console.WriteLine(); // Move to the next line after each row
28.            }
29.            Console.ReadKey();
30.        }
31.    }
32. }
```

```
01. using System;
02.
03. namespace ReversePyramidPattern
04. {
05.     class Program
06.     {
07.         static void Main(string[] args)
08.         {
09.             Console.WriteLine("iNetTutor.com - Reverse Pyramid");
10.             int height = 5;
11.
12.             // Outer loop iterates for each row
13.             for (int i = height; i >= 1; i--)
14.             {
15.                 // Inner loop prints spaces until reaching the current row number
16.                 for (int j = 1; j <= height - i; j++)
17.                 {
18.                     Console.Write(" ");
19.                 }
20.
21.                 // Inner loop prints stars for the current row
22.                 for (int k = 1; k <= 2 * i - 1; k++)
23.                 {
24.                     Console.Write("*");
25.                 }
26.
27.                 Console.WriteLine(); // Move to the next line after each row
28.             }
29.             Console.ReadKey();
30.         }
31.     }
32. }
```

## Explanation

This C# code generates a reverse pyramid pattern of stars with a height of 5. Here's a breakdown of its functionality:

- The `using System;` statement at the beginning of the code allows you to use classes and functions from the System namespace, such as Console for input and output operations [1](#).
- The code is enclosed within the `ReversePyramidPattern` namespace to organize the code and prevent naming conflicts with other code elements.
- The `Program` class contains the `Main` method, which serves as the entry point of the program.
- The `Console.WriteLine("iNetTutor.com - Reverse Pyramid");` statement prints a message indicating the purpose of the program.
- The `height` variable is set to 5, representing the desired height of the reverse pyramid.



iNetTutor.com

## Online Programming Lessons, Tutorials and Capstone Project guide

- The outer loop (for loop) iterates for each row of the pyramid, starting from the given height and decreasing by 1 in each iteration.
- The inner loop (for loop) prints spaces before the stars, where the number of spaces is determined by the difference between the height and the current row number.
- Another inner loop prints the stars for the current row, where the number of stars is calculated using the formula  $2 * i - 1$ .
- After printing each row, the `Console.WriteLine();` statement moves to the next line to start a new row.
- Finally, the `Console.ReadKey();` statement is used to prevent the console window from closing immediately after the program finishes execution, allowing you to view the output.

When you run this code, it will generate the desired reverse pyramid pattern of stars with a height of 5. Each row will have an increasing number of spaces before the stars, creating the inverted pyramid shape.

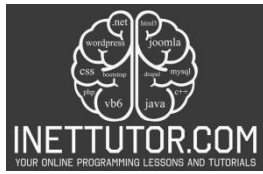
### Output

```
C:\Users\fujitsu\source\repos\ReversePyramidPattern\bin\Debug\  
iNetTutor.com - Reverse Pyramid  
*****  
  *****  
    *****  
      *****  
        *****
```

### Summary

In this lesson, we learned how to create a reverse pyramid pattern of stars using nested for loops in C#. The objectives of the lesson were to understand the concept of loops and nested loops, learn how to implement them to create patterns, practice writing code with nested loops, and apply the knowledge to generate a reverse pyramid pattern.

The provided C# source code demonstrated the implementation of nested for loops to generate the reverse pyramid pattern. The outer loop controlled the number of rows in the pyramid, while the inner



INetTutor.com

## Online Programming Lessons, Tutorials and Capstone Project guide

loops were responsible for printing spaces and stars in each row. By running the code, we were able to see the desired reverse pyramid pattern of stars with a height of 5.

Through this lesson, we gained a solid understanding of loop structures, enhanced our problem-solving skills, and developed the ability to create visually appealing patterns in C#. The knowledge and skills acquired in this lesson can be applied to various programming scenarios.

### Exercises and Assessment

Exercise:

1. Modify the existing code to allow the user to input the height of the reverse pyramid pattern. Prompt the user to enter the height and adjust the code accordingly to generate the pattern based on the user's input.
- 2.

Assessment:

1. Write a program that generates a reverse pyramid pattern of stars with a height of 7 using nested for loops. Ensure that the pattern is centered horizontally on the console.

Lab Exam with Rubrics: Objective: Create a program that generates a reverse pyramid pattern of stars using nested for loops and demonstrate understanding of loop structures and problem-solving skills.

Rubrics:

1. Correctness of the code (40%):
  - The program generates the reverse pyramid pattern correctly.
  - The pattern is centered horizontally on the console.
  - The height of the pyramid can be adjusted by the user.
2. Use of nested for loops (30%):
  - The program uses nested for loops to generate the pattern.
  - The outer loop controls the number of rows in the pyramid.
  - The inner loops handle the printing of spaces and stars in each row.
3. User input and interaction (15%):
  - The program prompts the user to enter the height of the pyramid.
  - The program adjusts the pattern based on the user's input.
4. Code organization and readability (15%):
  - The code is well-structured with appropriate indentation.
  - Variable names are meaningful and follow good naming conventions.
  - Comments are used to explain the logic and purpose of the code.

### Meta Description

Learn how to create a reverse pyramid pattern of stars using nested for loops in C#. Enhance problem-solving skills and create visually appealing patterns.