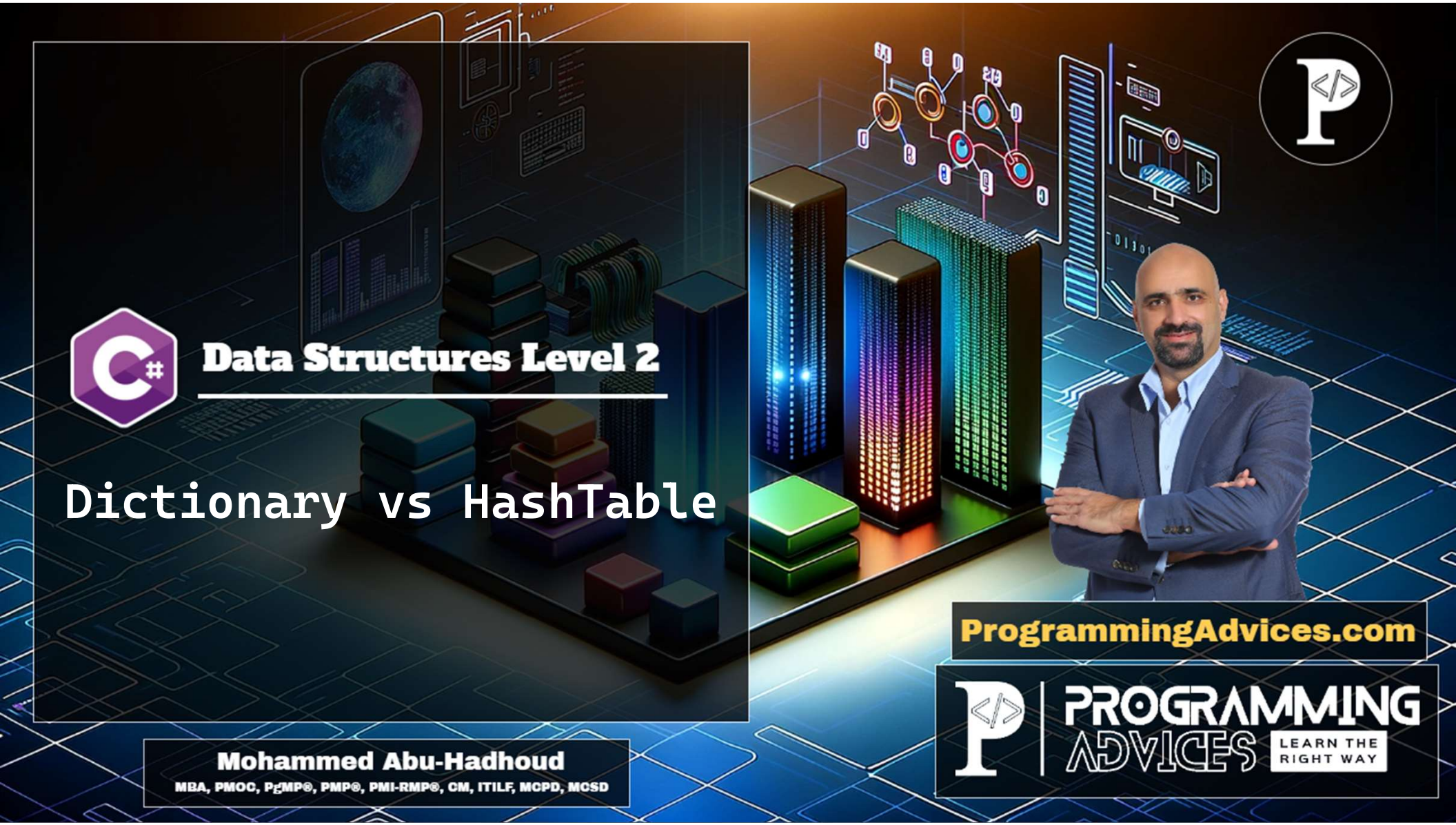

PROGRAMMING ADVICES
LEARN THE RIGHT WAY
26+ Years of Expericace
Mohammed Abu-Hadhoud
MBA, PMOC, PgMP®, PMP®, PMI-RMP®, CSE, ITILF, MCPD, MCSD
لا تنسى الاشتراك في قناتنا على اليوتيوب ومشاركة القناة مع اصدقائك
لتعم الفائدة للجميع وانقاذ الاف الناس من التشتت وجزاكم الله خيرا
لا تنسونا من دعائكم وادعو لوالدي بالرحمة
www.ProgrammingAdvices.com


مهم جداً
هذا الملف للمراجعة السريعة واخذ الملاحظات عليه فقط ،لانه يحتوي
على اقل من 20٪ مما يتم شرحه في الفيديوهات
الاستعجال والاعتماد عليه فقط سوف يجعلك تخسر كميه معلومات
وخبرات كثيره
يجب عليك مشاهدة فيديو الدرس كاملا
لاتنسى عمل لايك ومشاركة القناة لتعم الفائدة للجميع
لا تنسونا من دعائكم
ProgrammingAdvices.com
Mohammed Abu-Hadhoud
STOP
PROGRAMMING
ADVICES
LEARN THE
RIGHT WAY


Data Structures Level 2
Dictionary vs HashTable
Mohammed Abu-Hadhoud
MBA, PMOC, PgMP®, PMP®, PMI-RMP®, CM, ITILF, MCPD, MCSD
ProgrammingAdvices.com
PROGRAMMING ADVICES
LEARN THE RIGHT WAY

Dicitonary Vs HashTable?

- In C#, both Dictionary and Hashtable are collection types used to store key-value pairs. However, they are designed to cater to different needs and scenarios based on their features and implementations.
- Understanding the differences between Dictionary and Hashtable is crucial for choosing the appropriate collection type for a given situation.

Mohammed Abu-Hadhoud
MBA, PMOC, PgMP®, PMP®, PMI-RMP®, CM, ITILF, MCPD, MCSD
26+ years of experience

Dicitonary Vs HashTable

- Type:
  - Dictionary: Generic: Allows for type-safe data storage, ensuring that both keys and values are of a specified type, which helps to prevent runtime errors and eliminates the need for casting when retrieving values.
  - HashTable: Non-Generic: Keys and values are of type object, which means they can store any data type. This flexibility comes at the cost of type safety, as it requires casting when retrieving values and increases the chance of runtime errors.
- Performance:
  - Dictionary : Offers fast access to elements based on keys. The performance of searching for a key is close to O(1), making it highly efficient for lookups.
  - HashTable : also provides fast access to elements. However, the need for boxing and unboxing when working with value types can affect performance.

Mohammed Abu-Hadhoud
MBA, PMOC, PgMP®, PMP®, PMI-RMP®, CM, ITILF, MCPD, MCSD
26+ years of experience

Dicitonary Vs HashTable

- Order:
  - Dictionary: Does not guarantee the order of elements. The order in which elements are returned during enumeration may not match the order in which they were inserted.
  - HashTable: Does not maintain the order of stored elements, similar to Dictionary.
- Thread Safety:
  - Dictionary : Not inherently thread-safe. If multiple threads access it concurrently, you must implement your own synchronization mechanism.
  - HashTable : Provides some thread safety features, such as synchronized (thread-safe) wrappers obtained through the Hashtable.Synchronized method. However, for full thread safety with multiple writers, external synchronization is recommended.

Mohammed Abu-Hadhoud
MBA, PMOC, PgMP®, PMP®, PMI-RMP®, CM, ITILF, MCPD, MCSD
26+ years of experience

Dicitonary Vs HashTable

- UseCase:
  - Use Dictionary when you need strong type safety, better performance with value types, and are working with .NET 2.0 or later.
  - HashTable : Consider Hashtable if you are maintaining legacy code or need a collection that accepts keys and values of any type without specifying their data types upfront.

In modern .NET applications, Dictionary is generally preferred due to its type safety and performance advantages. However, understanding Hashtable is still valuable for working with existing codebases that use it.

Mohammed Abu-Hadhoud
MBA, PMOC, PgMP®, PMP®, PMI-RMP®, CM, ITILF, MCPD, MCSD
26+ years of experience


ProgrammingAdvices.com
Thank You
Mohammmed Abu-Hadhoud
26+ Years of Experience
MBA, PMOC, PgMP®, PMP®, PMI-RMP®, CM, ITILF, MCPD, MCSD
PROGRAMMING ADVICES
LEARN THE RIGHT WAY