



26+ Years  
of Experience

**PROGRAMMING  
ADVICES** LEARN THE  
RIGHT WAY

**Mohammed Abu-Hadhoud**

MSA, PMOC, PMP®, PRP®, PSE-ITP®, CS, ITIL, MCP®, MCD



لا تنسى الاشتراك في قناتنا على اليوتيوب ومشاركة القناة مع اصدقائك  
لتعم الفائدة للجميع وانقاذ الاف الناس من التشتت جزاكم الله خيرا

**لا تنسونا من دعائكم وادعو لوالدي بالرحمة**

**[www.ProgrammingAdvices.com](http://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



# Data Structures

## Level 1

Complexity of an Algorithm

Big O  
 $O(1)$

**Mohammed Abu-Hadhoud**

MBA, PMOC, PgMP®, PMP®, PMI-RMP®, CM, ITILF, MCPD, MCSD



**ProgrammingAdvises.com**



**PROGRAMMING  
ADVISES**

LEARN THE  
RIGHT WAY



# Data Structures

## Level 1

---

### Quick Review

**Mohammed Abu-Hadhoud**

MBA, PMOC, PgMP®, PMP®, PMI-RMP®, CM, ITILF, MCPD, MCSD



**ProgrammingAdvises.com**



Printing Array[10] Items  
Is faster than  
Printing Array[1000] Items

Program Speed

Algorithm

Hardware, CPU, RAM, OS

Performance

Time

Space

Complexity

Function (Input Size)

Time vs Input

Space vs Input

Input Size

$O(n)$



# Big O for the following Algorithms:

Algorithm 1

```
char GetLastCharacter(string S1)
{
    return S1[ S1.length() - 1 ];
}
```

Algorithm 2

```
char GetLastCharacter2(string S1)
{
    int n = S1.length() - 1 ;

    for ( int i = 0 ; i <= n ; i++ )
    {
        if ( i == n )
        {
            return S1 [ n ] ;
        }
    }
}
```



# Data Structures

## Level 1

# Calculating Algorithm Complexity $O(1)$

**Mohammed Abu-Hadhoud**

MBA, PMOC, PgMP®, PMP®, PMI-RMP®, CM, ITILF, MCPD, MCSD



**ProgrammingAdvises.com**



**PROGRAMMING  
ADVISES**

LEARN THE  
RIGHT WAY

# Calculating Algorithm Complexity:

- Count the number of steps during execution.
- Simplification:
  - Each step costs the same time.
  - +,-,return, access array element, ... etc.

```
char GetLastCharacter(string S1)
```

```
{  
    return S1[S1.length() - 1];  
}
```

Big O =  $4 * O(1) = 4 O(1)$   
Remove Factor  
 $O(1)$

- Number of Steps: 4 Steps, and it will be always 4 steps, it has nothing to do with array size ☺
- Constant Time , independent of array size

