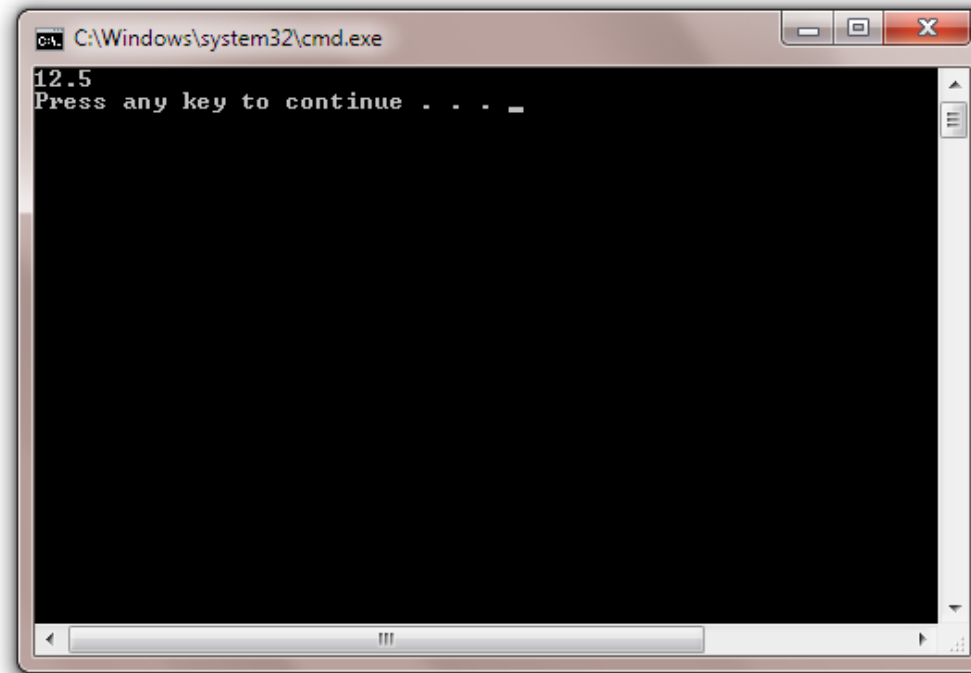


# متد WriteLine()

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(25/2.0);
        }
    }
}
```

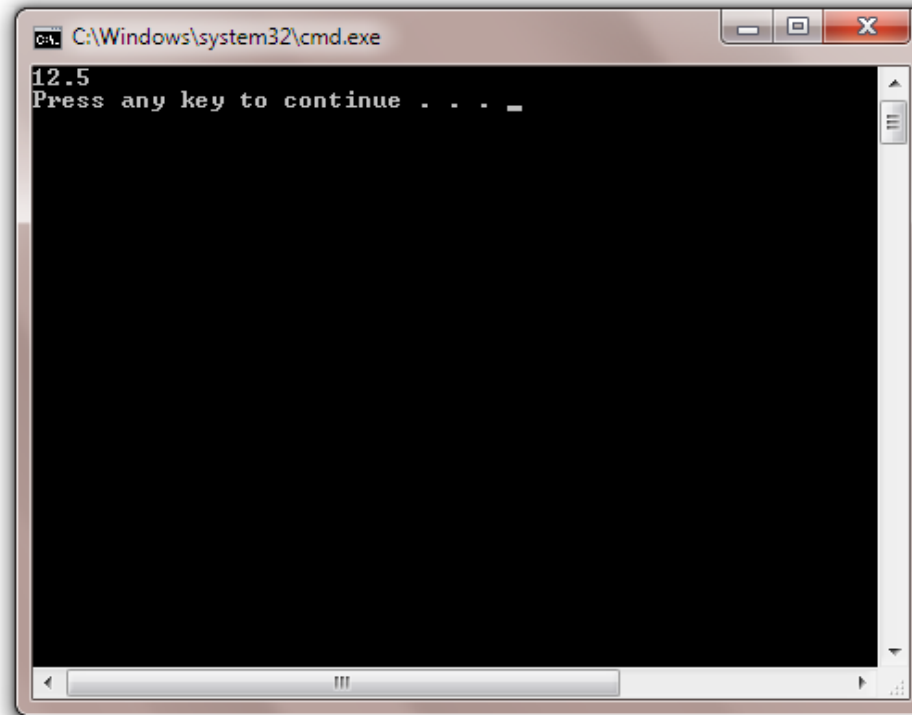


The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window displays the output of the program: "12.5" on the first line and "Press any key to continue . . . \_" on the second line. The cursor is positioned at the end of the second line.

# متد WriteLine()

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(25.0/2);
        }
    }
}
```



## کاراکتر بک اسلش (\)

عملکرد	دنباله
ایجاد یک بوق هشدار <sup>۲</sup>	\a
حذف یک کاراکتر (Backspace)	\b
ایجاد یک خط خالی (New Line)	\n
ایجاد یک فاصله افقی زیاد tab	\t
ایجاد یک تک کوتیشن ('')	\'
ایجاد یک دابل کوتیشن ("")	\"
ایجاد یک Back slash (\)	\\

❖ هر گاه کاراکتر بک اسلش در **یک رشته** دیده شود، کاراکتر بعدی آن اثر و عملکرد خاصی از خود نشان می دهد.

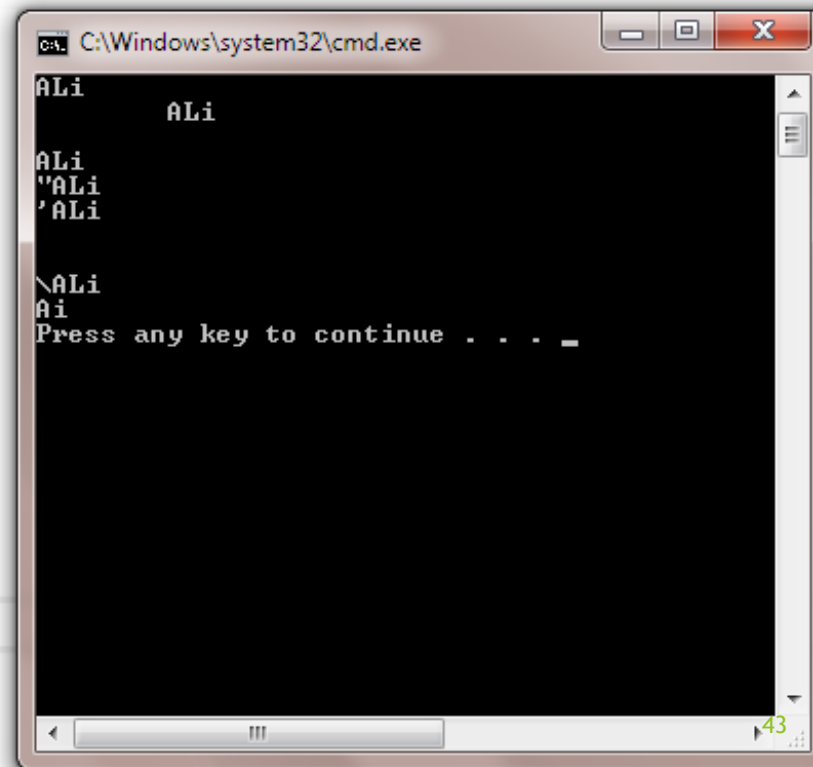
❖ به همین دلیل به این مجموعه کاراکترها که کاراکتر اول بک اسلش (\) و دومی کاراکتر دیگری است، **دنباله فرار** می گویند.

❖ به عنوان مثال در **ایجاد یک خط خالی** کاراکتر اول \ و کاراکتر بعدی n می باشد.

جدول - نحوه استفاده از کاراکتر بک اسلش (\)

# کاراکتر بک اسلش (\)

```
namespace Introduction
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("ALi");
            Console.WriteLine("\tALi");
            Console.WriteLine("\nALi");
            Console.WriteLine("\"ALi");
            Console.WriteLine("'ALi");
            Console.WriteLine("\n");
            Console.WriteLine("\\ALi");
            Console.WriteLine("AL\b");
        }
    }
}
mohsen.rezghjoo@modares.ac.ir
```



```
C:\Windows\system32\cmd.exe
ALi
    ALi
ALi
'ALi
'ALi
\nALi
Ai
Press any key to continue . . . _
```

# عمل الحاق رشته ها

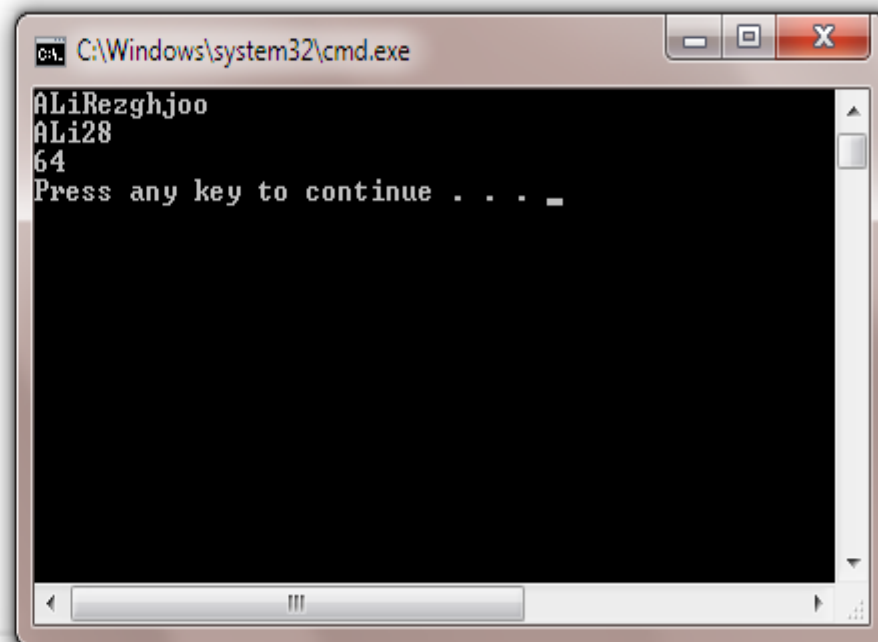
▶ هنگامی که علامت + حداقل با یک رشته به کار رود عمل الحاق رشته ها را انجام می دهد.

```
namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("ALi"+"Rezghjoo");

            Console.WriteLine("ALi" + 28);

            Console.WriteLine(48 + 16);
        }
    }
}
```

mohsen.fezghjoo@modares.ac.ir

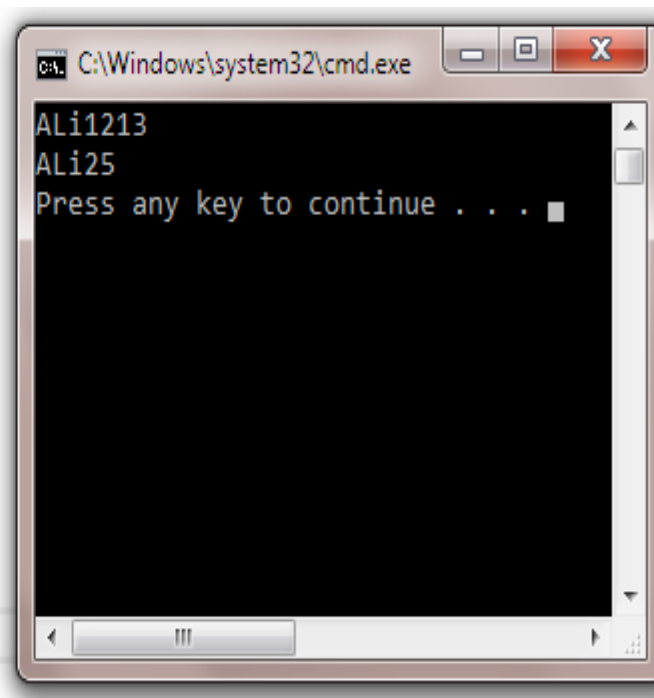


The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is displayed as follows:

```
ALiRezghjoo
ALi28
64
Press any key to continue . . . _
```

# عمل الحاق رشته ها

```
{  
class Program  
{  
    static void Main(string[] args)  
    {  
        Console.WriteLine("ALi" + 12 + 13);  
        Console.WriteLine("ALi" + (12 + 13));  
    }  
}
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is displayed on a black background with white text: "ALi1213", "ALi25", and "Press any key to continue . . .".

# الگوی جای گذاری در رشته

- ▶ به جای استفاده از عملگر + می توان در متد WriteLine() از الگوی جای گذاری مطابق روش زیر استفاده کرد.
- ▶ در این روش به جای {} مقدار متناظرشان قرار می گیرد. (اعداد داخل اکولاد از صفر شروع می شوند)
- ▶ همچنین می توان از متغیر هم استفاده کرد.
- ▶ به عنوان مثال اگر در مثال زیر {3} نوشته شود در مرحله اجرا خطا رخ می دهد.

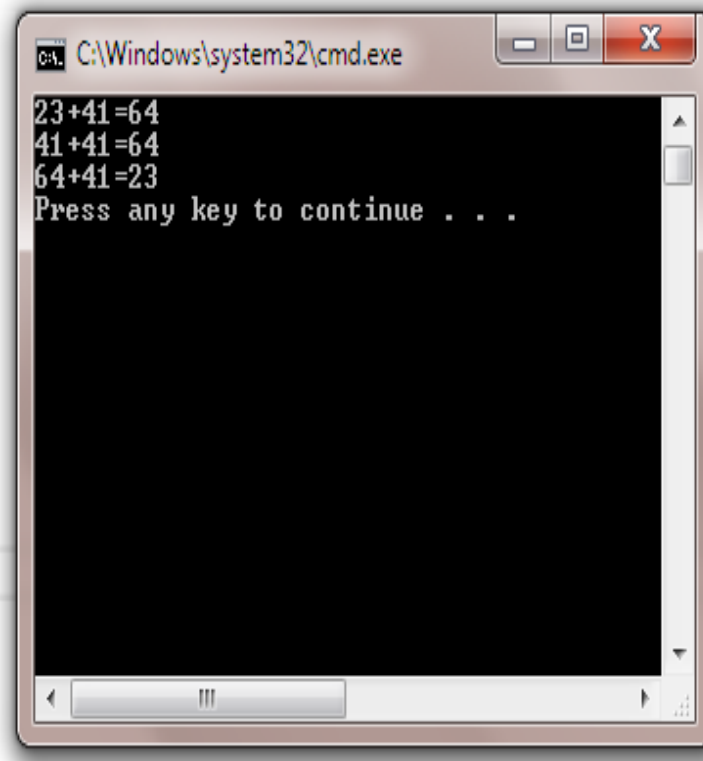
```
Console.WriteLine("{0} plus {1} is equal to {2}", 17, 45, 17+45);
```

# الگوی جای گذاری در رشته

```
{
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("{0}+{1}={2}",23,41,23 + 41);

        Console.WriteLine("{1}+{1}={2}",23,41,23 + 41);

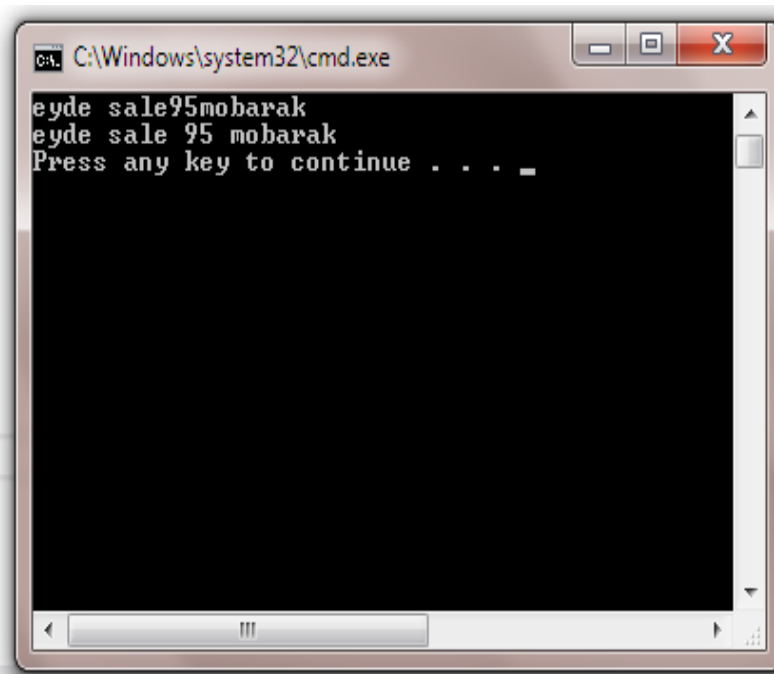
        Console.WriteLine("{2}+{1}={0}",23,41,23 + 41);
    }
}
```





# الگوی جای گذاری در رشته

```
{
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("eyde sale"+ 95 +"mobarak");
        Console.WriteLine("eyde sale {0} mobarak",95);
    }
}
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window contains the following text:

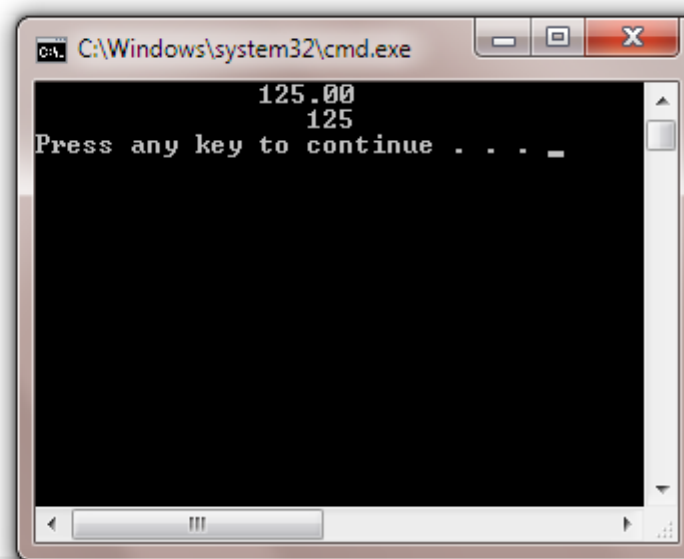
```
eyde sale95mobarak
eyde sale 95 mobarak
Press any key to continue . . . .
```

# الگوی جای گذاری در رشته

- ▶ علاوه بر محل قرارگیری یک عبارت یا یک متغیر در یک رشته، می توانیم الگو و طریقه نمایش عبارت و همچنین تراز آن را در صورت نیاز معین کنیم.

{الگوی نمایش : عدد تراز، شماره}

```
{
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("{0,20:F}",125);
        Console.WriteLine("{0,20}",125);
    }
}
```



# افزودن توضیحات در برنامه

## نکته

برای درج توضیحات در برنامه، اگر یک خط باشد از علامت // و چنانچه چند خط باشد از علامت /\* توضیحات \*/ استفاده کنید:

```
// Display a greeting message
```

```
/*
```

```
    FileName: welcome.cs ... Date : 05_07_2014
```

```
    Display a greeting message
```

```
*/
```

# افزودن توضیحات در برنامه

```
class WelcomeToCSharp
{
static void Main()
{
    System.Console.WriteLine("Welcome To C#!");
    // Insert a blank line
    System.Console.WriteLine();
    System.Console.WriteLine("This is my first program.");
}
}
```

# متد () Beep

- ▶ از این متد که در کلاس کنسول تعریف می شود برای ایجاد یک صدا یا صوت در برنامه استفاده می شود.
- ▶ اگر این متد به صورت خالی یعنی `Console.Beep()` به کار رود یک صوت به مدت یک ثانیه ایجاد می کند.
- ▶ اگر بخواهیم مدت زمان نواختن صدا و همچنین فرکانس (زیر و بم) صدا را تغییر دهیم کافی است در داخل پارانتز اعداد را به ترتیب بنویسیم:

(مدت زمان برحسب میلی ثانیه, فرکانس بر حسب هرتز) `Console.Beep`;

- ▶ برای تبدیل ثانیه به میلی ثانیه آن را در 1000 ضرب می کنیم یعنی دو ثانیه همان 2000 میلی ثانیه است.
- ▶ گوش انسان می تواند فرکانس حدود 200 تا 10000 هرتز را خوب بشنود.

# متد `ReadKey()` و `Clear()`

- ▶ متد `ReadKey()` که از کلاس کنسول می باشد منتظر دریافت یک کلید از روی صفحه کلید است یعنی کامپیوتر منتظر زدن کلیدی توسط کاربر باقی می ماند.
- ▶ از متد `Console.Clear();` برای پاک کردن صفحه نمایش پنجره کنسول در سی شارپ استفاده می شود.

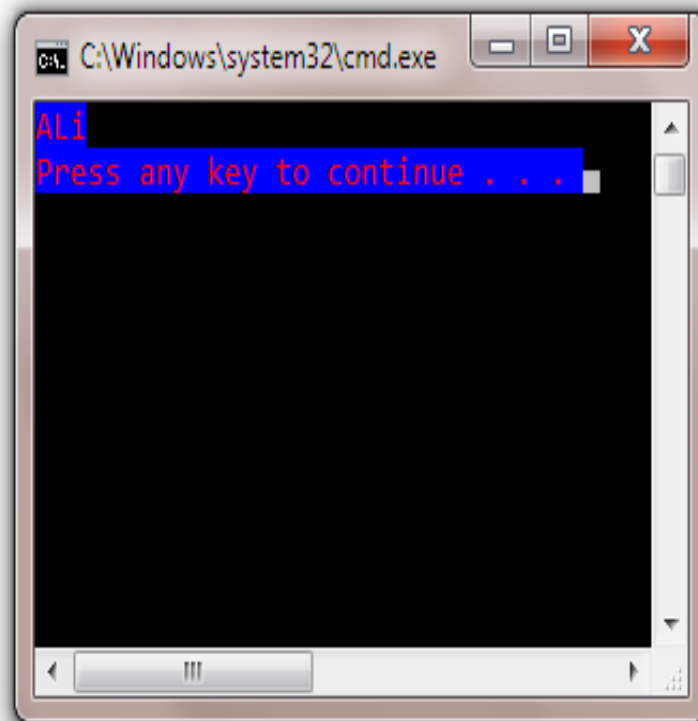
# متد رنگ

```
static void Main(string[] args)  
{
```

```
    Console.BackgroundColor = ConsoleColor.Blue;
```

```
    Console.ForegroundColor = ConsoleColor.Red;
```

```
    Console.WriteLine("ALi");
```



# متد رنگ

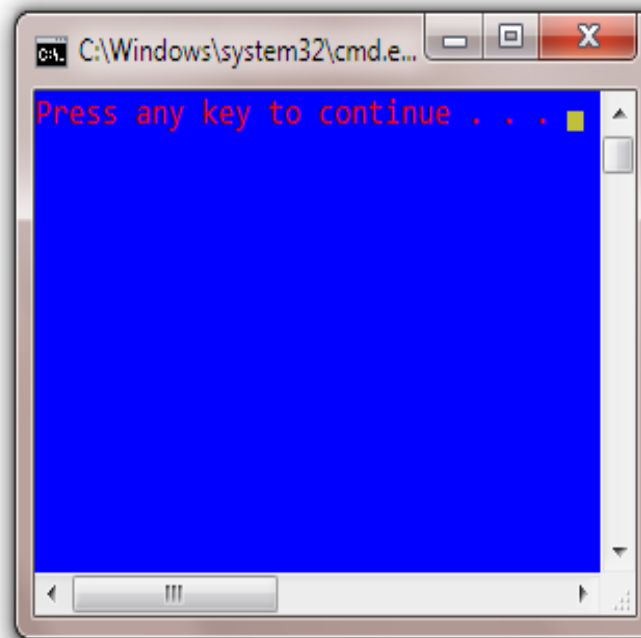
```
static void Main(string[] args)
{

    Console.BackgroundColor = ConsoleColor.Blue;

    Console.ForegroundColor = ConsoleColor.Red;

    Console.WriteLine("ALi");

    Console.Clear();
```





پیرم و گاهی دلم یاد جوانی می‌کند  
بلبلک شوقم هوای نغمه جوانی می‌کند

همت‌م تا می‌رود سار غزل کپرد به دست  
طاقتم اظهار عجز و ناتوانی می‌کند

بلبلی در سینه می‌نالد هنوزم کاین چمن  
با خزان هم آشتی و کل‌فشانی می‌کند

ما به داغ عشقباری‌ها نشستیم و هنوز  
چشم پروین همچنان چشمک‌پرانی می‌کند

نای ما خاموشی ولی این زهره‌ی شیطان هنوز  
با همان شور و نور دارد شبانی می‌کند

گر زمین بود هوا کرد همانا آسمان  
با همین ثبوت که دارد آسمانی می‌کند

سالها شد رفته همسارم ز دست اما هنوز  
در درونم زنده است و زندگانی می‌کند

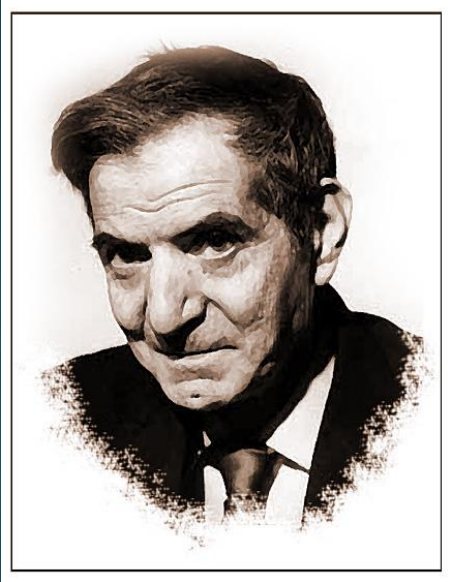
با همه نسیان تو کوی کز پی آزار من  
خاطرم با خاطرات خود تباری می‌کند

بی‌نمر هر ساله در فکر بهارتم ولی  
چون بهاران می‌رسد با من خزان می‌کند

طفل بودم در دمی پیر و علیم ساختند  
آنچه کردند می‌کند با ما نهانی می‌کند

می‌رسد قرنی به پایان و سپهر بایگان  
دقتر دوران ما هم بایگانی می‌کند

شهریارا گو دل از ما مهربانان مشکند  
ور نه قاضی در قضا نامهربانی می‌کند



## فصل چهارم

# آشنایی با انواع داده ها و متغیرها (Variable)



# متغیر (Variable)

- ▶ متغیر مکانی از حافظه RAM کامپیوتر است که برای نگهداری **موقتی** داده ها یا اطلاعات استفاده می شود.
- ▶ در یک برنامه برای نگهداری هر یک از داده ها با توجه به نوع و بزرگی داده، باید از متغیر مناسبی استفاده کنیم که بتوانیم داده را نگهداری کنیم.

## روش تعریف (اعلان) و ایجاد متغیرها:

نوع داده همان نوع متغیر است.  
دو روش برای تعریف متغیر وجود دارد.

2

مقدار = نام متغیر نوع داده

```
byte age = 16 ;
```

1

نام متغیر نوع داده

مقدار = نام متغیر

```
byte age ;
```

```
age = 16 ;
```

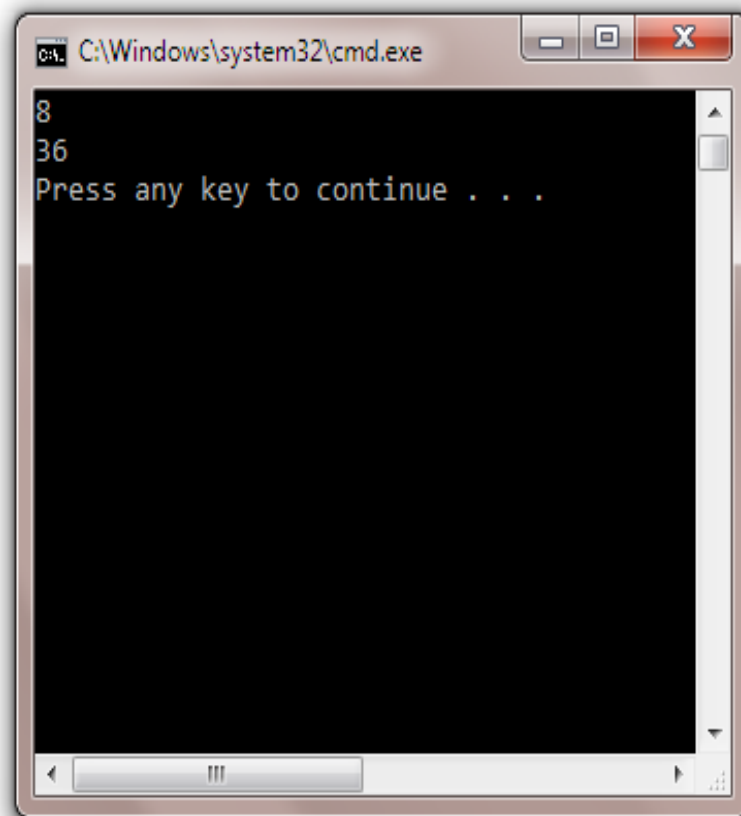
# متغیر (Variable)

- ▶ در یک متغیر همواره **فقط یک مقدار** نگهداری می شود و با ذخیره کردن داده جدید در یک متغیر، مقدار قبلی آن از بین می رود.
- ▶ در روش اول موجود در اسلاید قبل، اول متغیر تعریف و سپس مقداری در متغیر قرار می گیرد که به این دستور، **دستور انتساب (Assignment)** می گویند.
- ▶ در روش دوم موجود در اسلاید قبل، در هنگام تعریف یا ایجاد متغیر آن را مستقیماً مقداردهی کرده ایم که به آن **مقداردهی اولیه (Initialize)** می گویند.
- ▶ نام متغیر توسط برنامه نویس و طبق قواعد خاص نوشته می شود و منظور از نام متغیر یعنی ظرفی ایجاد می شود که می تواند فقط یک مقدار (با توجه به نوع متغیر) در خود ذخیره کند.
- ▶ اگر متغیر مقدار دهی نشود خطا میگیرد و همچنین اول باید متغیر را تعریف کرد و سپس مقداری در آن قرار داد.
- ▶ اگر بخواهید چند متغیر را از یک نوع تعریف کنید می توانید بعد از ذکر نوع داده، نام متغیرها را با علامت ویرگول از یکدیگر جدا کنید به مثال زیر توجه کنید :

در یک متغیر همواره **فقط یک مقدار** نگهداری می شود و با ذخیره کردن داده جدید در یک متغیر، مقدار قبلی آن از بین می رود.

```
static void Main(string[] args)
{
    int a, b;
    a = 5;
    b = 24;
    a = 8;
    b = 36;

    Console.WriteLine(a);
    Console.WriteLine(b);
}
```



# قواعد نام گذاری نام متغیر

- ▶ در زبان سی شارپ در نام گذاری نام متغیرها، رعایت موارد زیر الزامی است :
  1. استفاده از حروف الفبا، اعداد و کاراکتر زیر خط، مجاز است.
  2. نام متغیر نمی تواند با عدد شروع شود.
  3. نام انتخابی نمی تواند با کلمات کلیدی و یا رزرو شده باشد.
  4. استفاده از علامت فاصله و خط تیره و علائم دیگر ( @!+&%^#\$\* ) ( - ) مجاز نیست.



- ▶ روش های عمومی مورد استفاده در نام گذاری نام متغیر :
  - ❖ روش پاسکال : `AliRezghjoo` و `UserName`
  - ❖ روش کوهان شتری : `aliRezghjoo` و `userName`
  - ❖ روش مجارستانی : `intnumber` و `flonumber`

▶ با توجه به حساسیت زبان سی شارپ به حروف کوچک و بزرگ، در نام گذاری متغیرها دقت کنید که متغیر `a` و `A` مستقل هستند.

# قواعد نام گذاری نام متغیر

نام متغیر	توضیح
1a	غیر مجاز، نام متغیر نباید با عدد شروع شود.
a1	مجاز
employee Salary	غیر مجاز، بین کلمات نباید فاصله وجود داشته باشد
First	مجاز
Hello!	غیر مجاز، علامت تعجب نباید در نام وجود داشته باشد
payRate	مجاز
one+two	غیر مجاز، علامت + در یک نام نباید قرار داشته باشد
Conversion	مجاز
counter_1	مجاز
2nd	غیر مجاز، نام نمی تواند با عدد شروع شود

# نوع داده (نوع متغیر)

## ▶ نوع متغیر به طور کلی 3 ویژگی را مشخص می کند:

- (1) گنجایش یا ظرفیت متغیر : مثلا نوع int چهار بایت حافظه را اشغال می کند.
- (2) نوع اطلاعاتی که در متغیر می توان ذخیره کرد.
- (3) چه عملیاتی را می توان بر روی آن انجام داد.

- ❖ به عنوان مثال نوع داده sbyte اعداد صحیح (بدون اعشار) را در محدوده -128 تا +127 شامل می شود و فقط یک بایت حافظه را اشغال می کند. یعنی عدد 78 را میتوان در این نوع داده ذخیره کرد اما عدد 200 را نمی توان ذخیره کرد(کامپایلر خطا می گیرد)
- ❖ بنابراین اگر عددی یا حرف یا کاراکتری را بخواهیم در داخل یک متغیر ذخیره کنیم که خارج از ظرفیت و گنجایش آن متغیر باشد، مترجم متوجه آن شده و اجازه نمی دهد.
- ❖ **در جدول اسلاید صفحه بعد انواع داده آمده است.**



نوع داده	کاربرد نوع داده	مقدار حافظه (بایت)	کمترین مقدار	بیشترین مقدار
sbyte	اعداد صحیح	1	-128	127
byte	اعداد صحیح مثبت	1	0	255
short	اعداد صحیح	2	-32768	32767
ushort	اعداد صحیح مثبت	2	0	65535
int	اعداد صحیح	4	-2147483648	2147483647
uint	اعداد صحیح مثبت	4	0	4294967295
long	اعداد صحیح	8	-9223372036854778508	9223372036854778507
ulong	اعداد صحیح مثبت	8	0	18446744073709551615
float	اعداد اعشاری	4	$-3.402823 \times 10^{38}$	$3.402823 \times 10^{38}$
double	اعداد اعشاری با دقت زیاد	8	$-1.79769313486232 \times 10^{308}$	$1.79769313486232 \times 10^{308}$
decimal	اعداد صحیح بزرگ اعداد اعشاری با دقت بسیار زیاد	16	$-79228162514264337593543950335$ $-7.9 \times 10^{28}$	$79228162514264337593543950335$ $+7.9 \times 10^{28}$
bool	مقدار منطقی	1	false	true
char	یک حرف یا علامت (کراکتر)	2	0 کد کراکتر مطابق با سیستم Unicode	65535 کد کراکتر مطابق با سیستم Unicode
string	رشته			
object	آدرس یک داده			

عددی

غیر عددی

# نوع داده (نوع متغیر)

- ▶ اگر در اول نوع داده **U** بیاید اعداد منفی را شامل نمی شود. (از صفر شروع می شود به بالا)
- ▶ برای اعداد اعشاری می توانیم از نوع داده های `float` ، `double` و `decimal` استفاده کنیم.
- ▶ نوع داده `float` برای اعداد اعشاری با دقت حداکثر 7 رقم استفاده می شود.
- ▶ نوع داده `double` برای اعداد اعشاری با دقت حداکثر 15 رقم استفاده می شود.
- ▶ اشغال فضای حافظه `double` (8بایت) دو برابر نوع `float` (4بایت) می باشد بنابراین برای ذخیره نمره یک درس متغیر نوع `float` مناسب است.
- ▶ هنگام تعریف متغیر از نوع `float` اخر عدد `F` یا `f` قرار میدهم ولی هنگام تعریف متغیر از نوع `double` نیازی به نوشتن `D` یا `d` نیست. (علت در اسلاید صفحه بعد)

# نوع داده (نوع متغیر)

## نکته

در زبان C# هر عدد اعشاری داخل برنامه، به وسیله مترجم به عنوان نوع double در نظر گرفته می شود. بنابراین اگر بخواهید یک عدد ممیزی را در یک متغیر نوع float ذخیره کنید مترجم خطا یا هشدار می دهد. برای جلوگیری از این مسئله باید از متغیرهای نوع double در هنگام کار با اعداد اعشاری استفاده کنید و یا اینکه در جلوی اعداد اعشاری حرف F یا f را بنویسید تا مترجم، این عدد را به عنوان یک عدد نوع float در نظر بگیرد.

```
double PI = 3.141592653589793238;
```

```
float myPhysicMark;  
myPhysicMark = 17.75f;
```

# تبدیل نوع

```
Program.cs*  X  
ConsoleApplication3.Program  Main(string[] args)  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        byte age = 16;  
        Console.WriteLine("Emsal sen man :" + age);  
        age = age + 1;  
        Console.WriteLine("Sale bad sen man :" + age);  
    }  
}
```

146 %

Error List - Open Documents

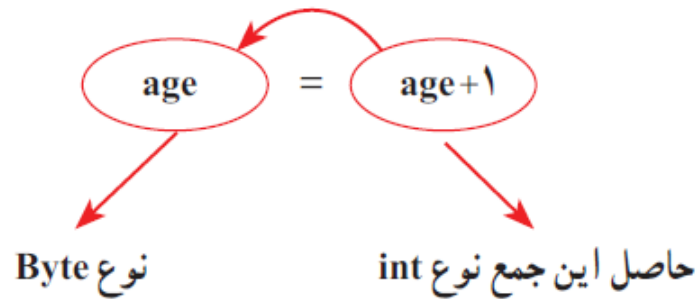
1 of 1 Error | 0 of 0 Warnings | 0 of 0 Messages

Description

1. Cannot implicitly convert type 'int' to 'byte'. An explicit conversion exists (are you missing a cast?)



# تبدیل نوع



- ▶ در عبارت  $age = age + 1$  عملگر  $+$  دارای دو عملوند است، یکی از عملوندها متغیر  $age$  از نوع  $byte$  (ظرفیت یک بایت) و عملوند دیگر عدد یک از نوع صحیح یا همان  $int$  (ظرفیت چهار بایت) است.
- ▶ ظرفیت داده  $int$  چهار بایت و ظرفیت داده  $byte$  یک بایت است بنابراین حاصل جمع عددی از نوع صحیح  $int$  خواهد بود و هنگامی که بخواهیم یک مقدار بزرگ ( $int$ ) را در یک مقدار کوچک ( $byte$ ) قرار دهیم مترجم خطا می گیرد.
- ▶ برای برطرف کردن اشکال، می توانیم از مترجم بخواهیم که حاصل جمع را به نوع  $byte$  تبدیل کند تا در متغیر  $age$  جای بگیرد. برای این منظور از تبدیل نوع استفاده می کنیم.
- ▶ نحوه استفاده از تبدیل نوع به شکل زیر است: (نکته مهم: اگر در عبارت مبدا بیش از یک عملگر یا متغیر باشد هر دو عبارت در داخل پارانترز نوشته شوند)

(عبارت مبدا) (نوع داده مقصد)

## مثال مربوط به تبدیل نوع

```
class Program
{
    static void Main(string[] args)
    {
        byte age = 16;
        Console.WriteLine("Emsal sen man :" + age);
        age = (byte) (age + 1);
        Console.WriteLine("Sale bad sen man :" + age);
    }
}
```



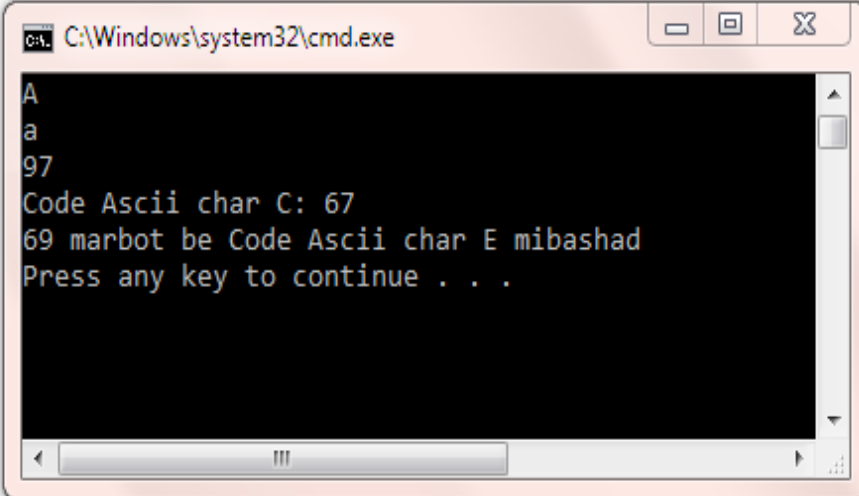
```
C:\Windows\system32\cmd.exe
Emsal sen man :16
Sale bad sen man :17
Press any key to continue . . .
```

# مثال مربوط به تبدیل نوع

```
class Program
{
    static void Main(string[] args)
    {
        int number = 65;
        Console.WriteLine((char)number);

        char ch = 'a';
        Console.WriteLine(ch);
        Console.WriteLine((int)ch);

        Console.WriteLine("Code Ascii char C: " + (int)'C');
        Console.WriteLine("69 marbot be Code Ascii char {0} mibashad ",(char)69);
    }
}
```

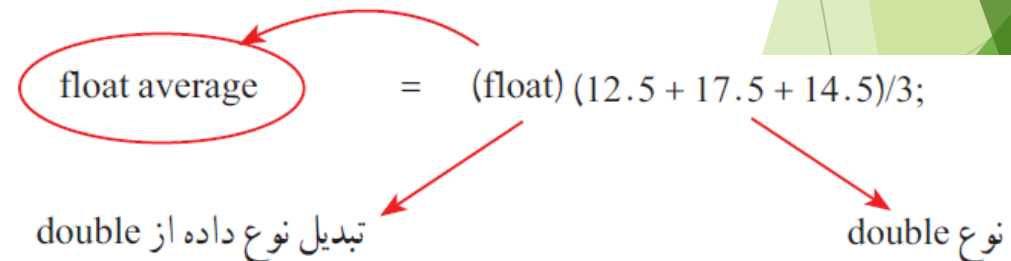
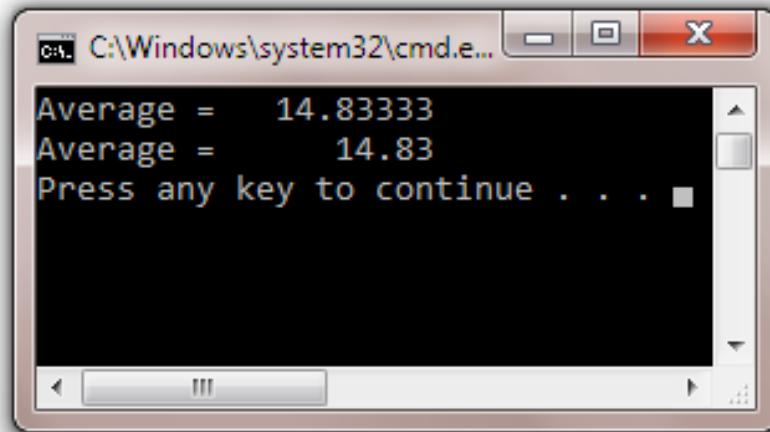


The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is as follows:

```
A
a
97
Code Ascii char C: 67
69 marbot be Code Ascii char E mibashad
Press any key to continue . . .
```

# مثال مربوط به تبدیل نوع

```
class Program
{
    static void Main(string[] args)
    {
        float average = (float)(12.5 + 17.5 + 14.5) / 3;
        Console.WriteLine("Average = {0,10}", average);
        Console.WriteLine("Average = {0,10:F}", average);
    }
}
```





## نکته:

- ▶ در یک برنامه به زبان سی شارپ می توانیم اعداد صحیح را در مبانی 16 نیز بنویسیم. برای این منظور قبل از عدد مورد نظر از پیشوند 0x و یا 0X استفاده می کنیم که نشان دهنده اعداد مبنا 16 می باشد.
- ▶ با اجرای دستور زیر در متغیر portValue مقدار 27 و در متغیر portaddress عدد 255 قرار می گیرد.

```
byte portValue = 0x1B;
```

```
ushort portAddress = 0X00FF;
```

$$(1B) = (1 \times 16) + 11 = 27$$

$$(FF) = (15 \times 16) + 15 = 255$$

## نحوه تعریف انواع متغیر

```
String a ;
```

```
a = "چندین کاراکتر" ;
```

```
char a ;
```

```
a = 'تنها یک کاراکتر' ;
```

```
bool a ;
```

```
a = false or true ;
```

```
int a ;
```

```
a = عدد ;
```

# نحوه تعریف انواع متغیر

## ▶ بولین (bool)

❖ نوع داده بولین فقط قادر است شامل دو مقدار درست (true) و نادرست (false) باشد، اگر مقدار دیگر قرار بگیرد کامپایلر خطا می گیرد.

## ▶ کاراکتر (char)

❖ کاراکتر عبارت است از یک حرف الفبا یا یک علامت و یا نشانه هایی مانند آنچه که در روی صفحه کلید قرار دارد یک کاراکتر می تواند با علامت آن در بین علائم تک کوتیشن (' ') قرار گیرد.

❖ در داده کاراکتری **تنها یک کاراکتر** باید بین علائم تک کوتیشن وجود داشته باشد.

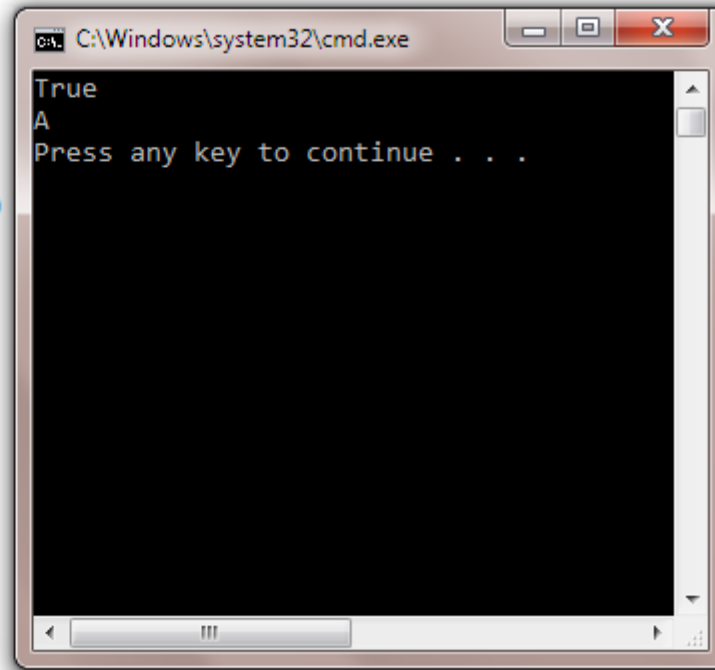
❖ در داخل تک کوتیشن فاصله نیز یک کاراکتر محسوب می شود.

❖ چند نمونه از کاراکترها را در زیر مشاهده می کنید:

mohsen.rezagholi@modares.ac.ir 'A' , 'a' , '&' , '\$' , '+' , ''

# نحوه تعریف انواع متغیر

```
namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {
            bool a = true;
            char b = 'A' ;
            Console.WriteLine(a);
            Console.WriteLine(b);
        }
    }
}
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is displayed on a black background with white text: "True", "A", and "Press any key to continue . . .".

# نحوه تعریف انواع متغیر

## ▶ رشته ای (String) :

- ▶ نوع داده کاراکتر تنها برای نگهداری یک کاراکتر مناسب است ولی برای هنگامی که نیاز به بیش از یک کاراکتر داشته باشیم (مثل نام خانوادگی) باید از نوع داده رشته ای استفاده کنیم.
- ▶ یک رشته شامل تعدادی حروف و کاراکتر است که در بین جفت کوتیشن (" ") قرار گرفته است.
- ▶ به عنوان مثال "mohammad" یک داده رشته ای است که شامل 8 کاراکتر است.
- ▶ در زیر یک مثال از رشته بیان شده است :

```
string name = "Mohammad";  
string message = "Welcome" + name;  
System.Console.WriteLine(message);
```

خروجی

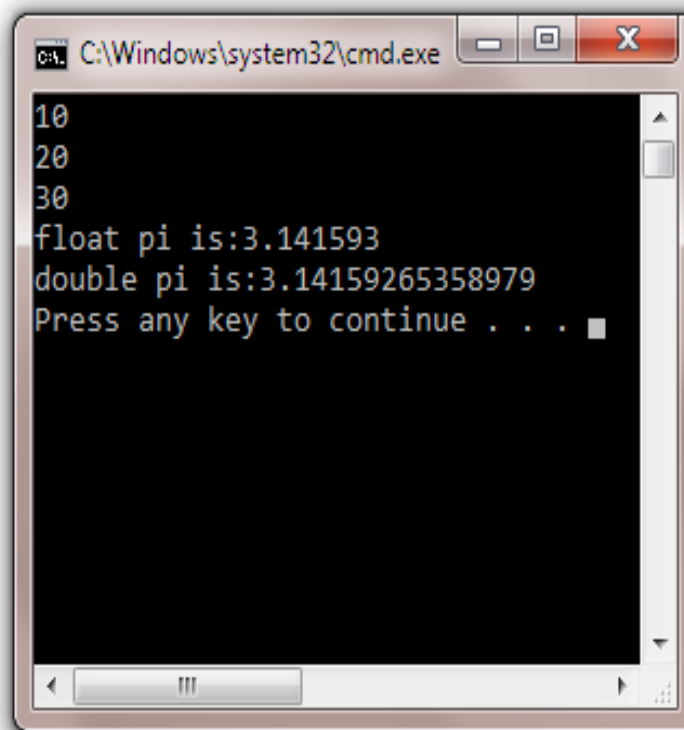
WelcomeMohammad

## نحوه تعریف انواع متغیر

```
static void Main(string[] args)
{
    int a = 10, b = 20, c;
    c = a + b;
    Console.WriteLine(a);
    Console.WriteLine(b);
    Console.WriteLine(c);

    float lowpi = 3.1415926535897932f;
    double highpi = 3.1415926535897932;

    Console.WriteLine("float pi is:" + lowpi);
    Console.WriteLine("double pi is:" + highpi);
}
```



```
C:\Windows\system32\cmd.exe
10
20
30
float pi is:3.141593
double pi is:3.14159265358979
Press any key to continue . . . ■
```

# کار با اعداد اعشاری

- ▶ در روش نماد علمی هر عدد از 2 بخش تشکیل شده است که با علامت ضرب از یکدیگر جدا می شوند. بخش اول یک عدد اعشاری بین 1 تا 9 که فقط یک رقم صحیح دارد که به این ها **مانتیس** می گویند و قسمت دوم به صورت توانی از عدد 10 است که به آن **نما** گفته می شود.
- ▶ در زبان سی شارپ از یک فرم نماد علمی برای نمایش اعداد اعشاری استفاده می شود که به آن **فرم نقطه شناور** گفته می شود. در این حالت توان 10 بعد از حرف E نوشته می شود.

فرم معمولی	فرم نماد علمی
4380000	$4.38 \times 10^6$
.0000265	$2.65 \times 10^{-5}$
47.9832	$4.79832 \times 10^1$
10000000	$1 \times 10^7$
-5600	$-5.6 \times 10^3$

عدد	نمایش عدد در فرم نقطه شناور
75.924	7.5924E1
0.18	1.8E-1
0.0000453	4.53E-5
-1.482	-1.482E0
7800.0	7.8E3