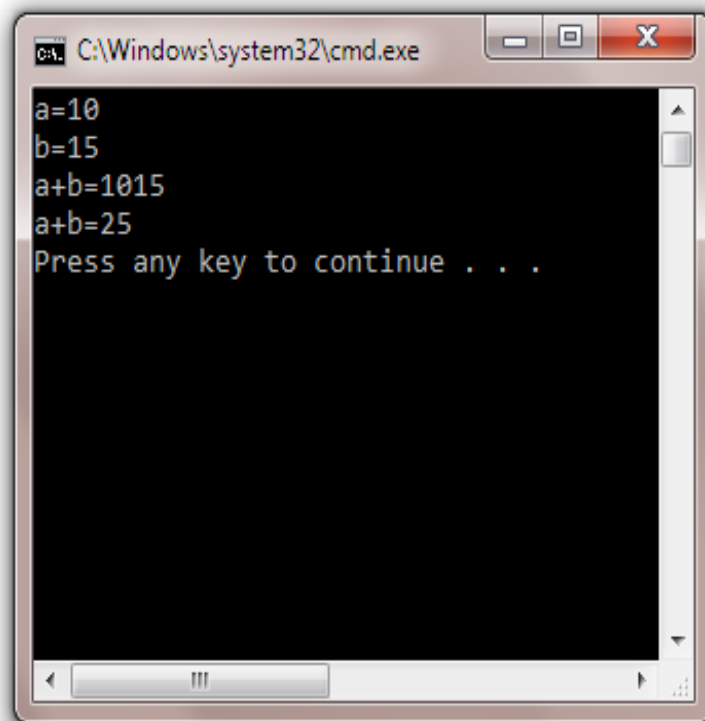


نحوه تعریف انواع متغیر

```
class Program
{
    static void Main(string[] args)
    {
        int a, b;
        a = 10;
        b = 15;

        Console.WriteLine("a=" + a);
        Console.WriteLine("b=" + b);

        Console.WriteLine("a+b=" + a+b);
        Console.WriteLine("a+b=" + (a+b));
    }
}
```



```
C:\Windows\system32\cmd.exe
a=10
b=15
a+b=1015
a+b=25
Press any key to continue . . .
```

دریافت رشته با متد ReadLine()

▶ برای این که داده ای از کاربر بتوانیم دریافت کنیم از متد ReadLine() از کلاس کنسول استفاده می کنیم. این متد به کاربر اجازه می دهد که داده مورد نظر خود را از طریق صفحه کلید وارد کند.

▶ کامپیوتر با اجرای این متد متوقف شده و منتظر دریافت داده از کاربر می شود. کاربر می تواند داده مورد نظر خود را تایپ کرده و در پایان دکمه Enter را بزند.

▶ داده ای که توسط کاربر وارد می شود به صورت یک رشته ذخیره می شود پس باید از یک متغیر رشته ای برای ذخیره داده دریافتی کاربر استفاده کنیم.

▶ بنابراین هنگام استفاده از دستور ReadLine() همیشه باید یک متغیر از نوع String تعریف کنیم و داده دریافتی توسط ReadLine() را در آن قرار دهیم.

▶ به عنوان مثال میخواهیم نام و نام خانوادگی

را از کاربر سوال کنیم:

```
string name, family;
```

```
name = System.Console.ReadLine();
```

```
family = System.Console.ReadLine();
```

دریافت رشته با متد ReadLine()

توجه داشته باشید وقتی کامپیوتر منتظر دریافت داده از کاربر است کاربر نیز باید بداند که چه داده ای را لازم است وارد کند بنابراین قبل از استفاده از متد ReadLine() یک دستور برای نمایش یک پیام و توضیحی کوتاه در مورد اینکه کامپیوتر منتظر چه داده ای است در برنامه باید نوشته شود. به مثال زیر توجه کنید :

```
string name ;
```

```
System.Console.WriteLine("Enter your name: ");
```

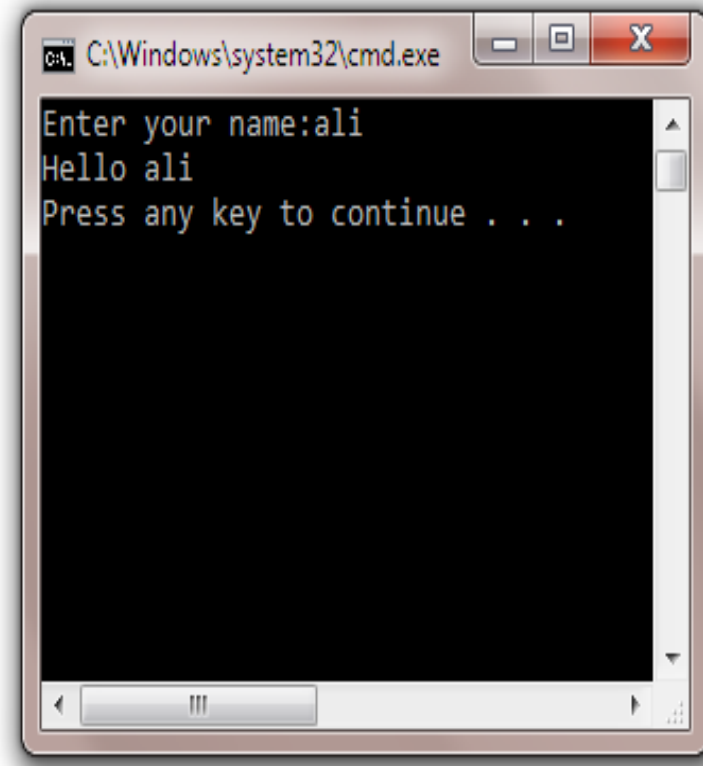
```
name = System.Console.ReadLine();
```

نکته

متد ReadLine() شبیه متد ReadKey() است با این تفاوت که متد ReadKey() فقط منتظر دریافت یک کلید می شود اما در متد ReadLine() تا هنگامی که کلید Enter زده نشده است کامپیوتر منتظر می ماند.

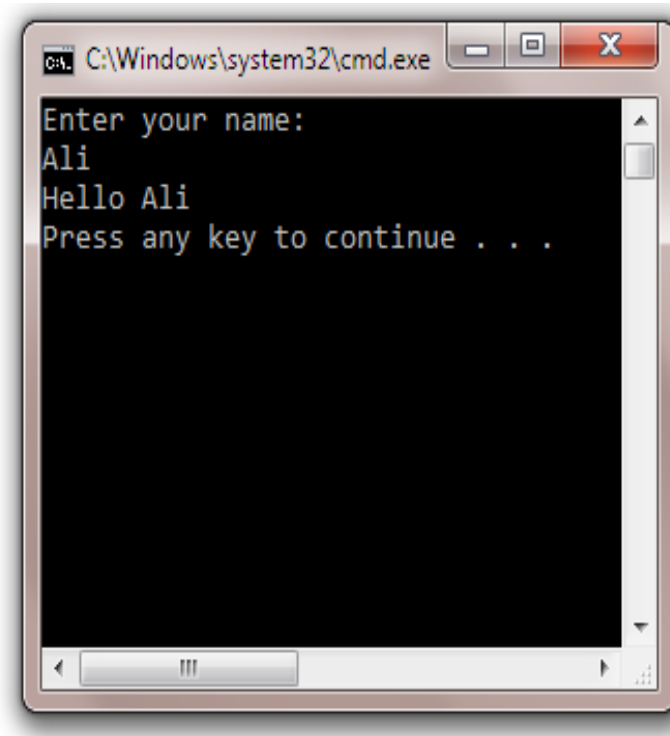
مثال: برنامه ای بنویسید که نام کاربر را از ورودی دریافت کرده و خطاب به او پیام خوشامدگویی اعلام کند؟

```
class Program
{
    static void Main(string[] args)
    {
        string name;
        Console.Write("Enter your name:");
        name = Console.ReadLine();
        Console.WriteLine("Hello" + " " + name);
    }
}
```



مثال: برنامه ای بنویسید که نام کاربر را از ورودی دریافت کرده و خطاب به او پیام خوشامدگویی اعلام کند؟

```
class Program
{
    static void Main(string[] args)
    {
        string name;
        Console.WriteLine("Enter your name:");
        name = Console.ReadLine();
        Console.WriteLine("Hello" + " " + name);
    }
}
```



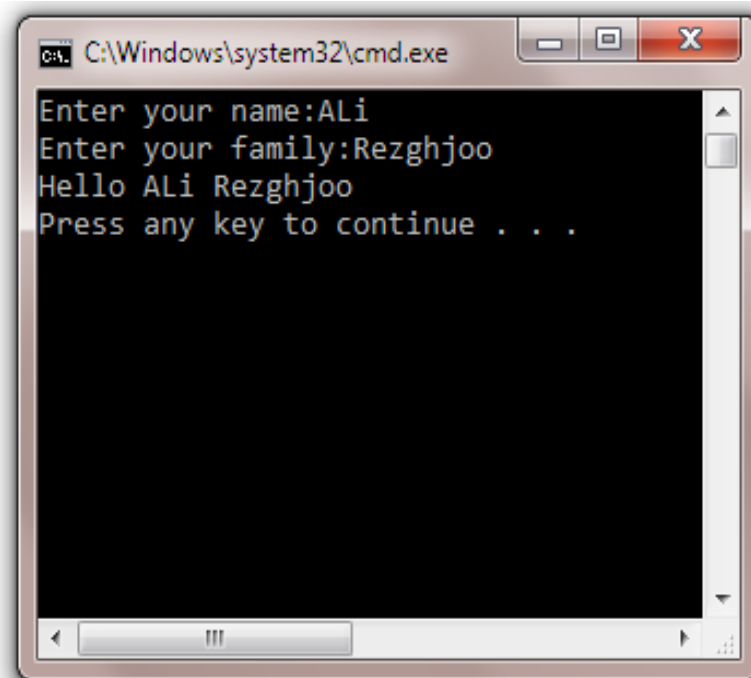
مثال: برنامه ای بنویسید که نام و نام خانوادگی کاربر را از ورودی دریافت کرده و خطاب به او پیام خوشامدگویی اعلام کند؟

```
static void Main(string[] args)
{
    string name, family;

    Console.Write("Enter your name:");
    name = Console.ReadLine();

    Console.Write("Enter your family:");
    family = Console.ReadLine();

    Console.WriteLine("Hello" + " " + name + " " + family);
}
```



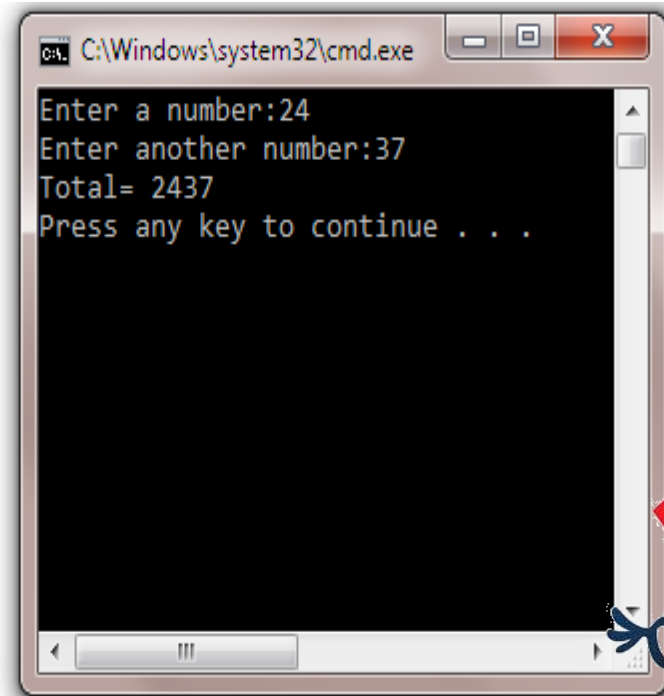
مثال: برنامه ای بنویسید که دو عدد دلخواه از کاربر دریافت کرده و مجموع آن ها را حساب و روی صفحه نمایش، نشان دهد؟

```
static void Main(string[] args)
{
    string firstNumber, secondNumber;

    Console.Write("Enter a number:");
    firstNumber = Console.ReadLine();

    Console.Write("Enter another number:");
    secondNumber = Console.ReadLine();

    Console.WriteLine("Total=" + " " + (firstNumber+secondNumber));
}
```



mohsen.rezghjoo@modares.ac.ir

پاسخ جواب : بر روی رشته ها نمی توان عملیات ریاضی مثل جمع انجام داد.

دریافت اعداد با متد Parse()

- ▶ با توجه با این که داده های دریافتی به وسیله متد ReadLine() همواره به صورت رشته می باشند باید به وسیله دستوری رشته دریافتی را بتوانیم به عدد تبدیل کنیم.
- ▶ متد Parse() که به آن تجزیه کردن می گویند می تواند یک رشته شامل ارقام را به معادل عددی آن تبدیل کند.
- ▶ به عمل بررسی کاراکتر به کاراکتر یک رشته، برای جدا کردن و بدست آوردن یک مقدار با معنی **تجزیه کردن** می گویند.
- ▶ مثلا برای تجزیه رشته "259" به ارزش عددی آن یعنی 259 ، چون داخل رشته عدد صحیح است از متد Parse() مربوط به نوع داده int استفاده می کنیم. حاصل این اجرا عدد 259 است که باید در یک متغیر نوع صحیح ذخیره شود.

Parse(); نوع متغیر

```
int.Parse("259");
```

```
int a ;  
a = int.Parse("259");
```

```
int a = int.Parse("259");
```


دریافت اعداد با متد Parse()

▶ با استفاده از متد Parse() می توانیم رشته دریافتی که به وسیله متد ReadLine() از کاربر دریافت شده را به عدد تبدیل کنیم.

```
string input;
```

```
float number;
```

```
input = Console.ReadLine();
```

```
number = float.Parse(input);
```

▶ همچنین متد ReadLine() را می توانیم مستقیماً در متد Parse() استفاده کرد که در این صورت نیازی به متغیر رشته ای نیست.

```
float number;
```

```
number = float.Parse(Console.ReadLine());
```

مثال: برنامه ای بنویسید که دو عدد دلخواه از کاربر دریافت کرده و مجموع آن ها را حساب و روی صفحه نمایش، نشان دهد؟

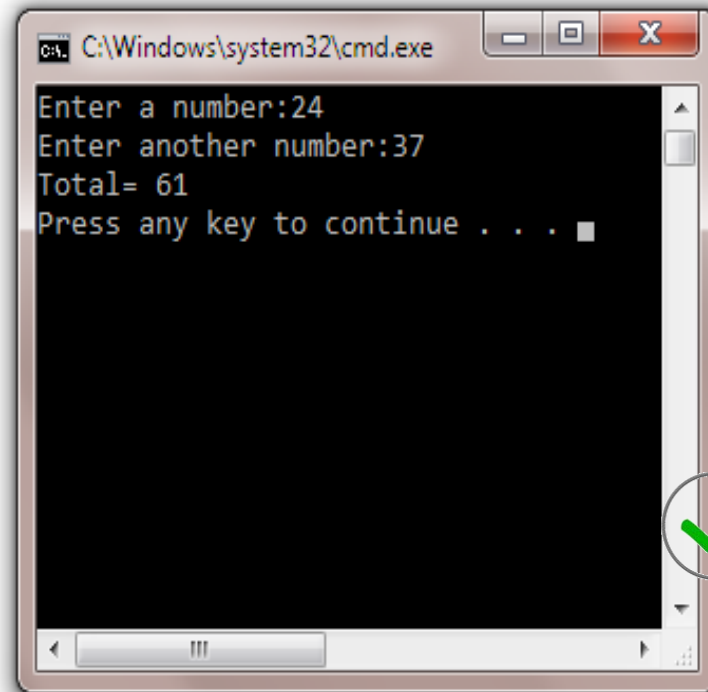
```
static void Main(string[] args)
{
    string input;
    float firstNumber, secondNumber;

    Console.Write("Enter a number:");
    input = Console.ReadLine();
    firstNumber = float.Parse(input);

    Console.Write("Enter another number:");
    input = Console.ReadLine();
    secondNumber = float.Parse(input);

    Console.WriteLine("Total=" + " " + (firstNumber+secondNumber));
}
```

mohsen.rezghjoo@modares.ac.ir



```
static void Main(string[] args)
{
    int a = 10, b = 20;

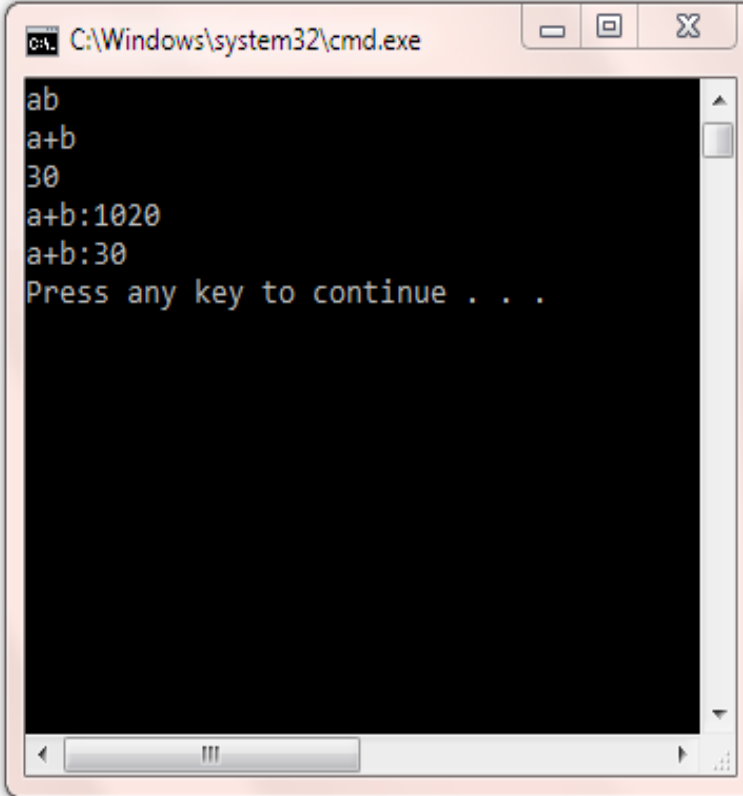
    Console.WriteLine("a"+"b");

    Console.WriteLine("a+b");

    Console.WriteLine(a+b);

    Console.WriteLine("a+b:"+a+b);

    Console.WriteLine("a+b:"+a+b);
}
```



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

```
ab
a+b
30
a+b:1020
a+b:30
Press any key to continue . . .
```

مردی مقابل گل فروشی ایستاده بود و میخواست دسته کلی برای مادرش که در شهر دیگری بود سفارش دهد تا برایش پست شود. وقتی از گل فروشی خارج شد، دفتری را دید که روی جدول خیابان نشسته بود و هق هق میکرد. مرد نزدیک رفت و از او پرسید: دفتر خوب چرا گریه میکنی؟ دفتر در حالی که گریه میکرد گفت: میخواستم برای مادرم یک شافه گل رز بفرم ولی..... فقط ۲۵ سنت دارم، در حالی که گل رز ۲ دلار میشود. مرد لبخند زد و گفت: با من بیا، من برای تو یک شافه گل رز قشنگ میفرم.



وقتی از گل فروشی خارج میشدند، مرد به دفتر گفت: مادرت کجاست؟ میفواهی ترا برسانم؟ دفتر دست مرد را گرفت و گفت: آنجا و به طرف قبرستان اشاره کرد.



مرد او را به قبرستان برد و دفتر روی یک قبر تازه نشست و گل را آنجا گذاشت. مرد دلش گرفت و طاقت نیاورد، به گل فروشی برگشت، دسته گل را گرفت و ۲۰۰ مایل رانندگی کرد تا خودش دسته گل را به مادرش بدهد.....
(قدر فرشتگان زمینی خود را بدانیم)



فصل پنجم

عبارت های محاسباتی



عبارت های محاسباتی

► **تعریف عملگر :**

در عبارت $4+2$ به علامت جمع عملگر می گویند. بنابراین به علامت هایی که بیانگر انجام یک عمل بر روی اعداد و داده ها هستند، عملگر گفته می شود.

► **تعریف عملوند :**

به اعداد 4 و 2 موجود در عبارت $2+4$ عملوند می گویند. بنابراین به اعدادی که یک عملگر بر روی آنها عملی را انجام می دهد عملوند می گویند.

► هر یک از عملگرهای ضرب و جمع بر روی دو عدد عمل می کنند به عبارتی دارای دو عملوند هستند به این عملگرها ، **عملگرهای دوتایی** گفته می شود.

► عملگر قرینه (-3) فقط دارای یک عملوند است به همین خاطر به آن **عملگر یکتایی** می گویند.

► اگر در عبارتی بیش از یک عملگر وجود داشته باشد از چپ به راست عملگری که اولویت بالاتری دارد به ترتیب انجام می شوند که به آن «**شرکت پذیری چپ**» می گویند.

► اولویت عملگرهای ریاضی در اسلاید بعد آمده است ولی اگر در عبارت پارانتز وجود داشته باشد اولویت آن از همه بالاتر است و اگر چند پارانتز تودرتو باشد ابتدا داخلی ترین پارانتز انجام می شود.

عبارت های محاسباتی

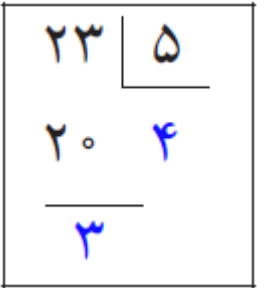
اولویت	نام عملگر	نشانه	مثال	نوع عملگر
۱	قرینه	-	-5	یکتایی
۲	ضرب	*	$12 * 36$	دو تایی
	تقسیم	/	$25/4$	
۳	باقیمانده تقسیم	%	$23 \% 5$	دو تایی
	جمع	+	$75 + 14$	
	تفریق	-	$29 - 36$	

جدول - اولویت عملگرهای ریاضی

عبارت های محاسباتی

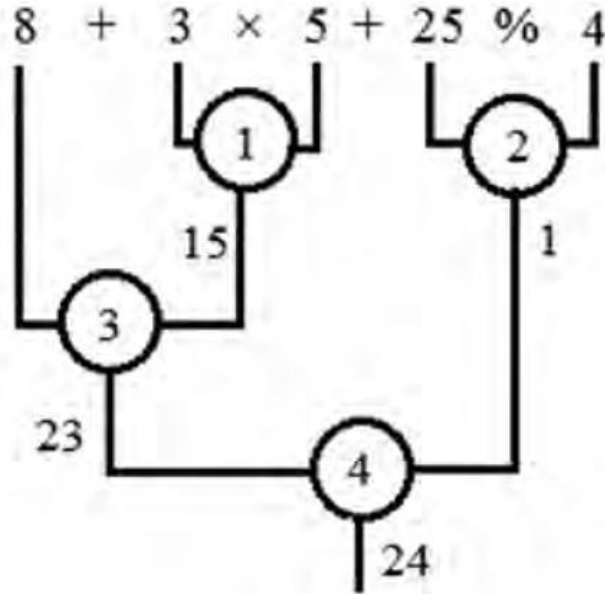
عبارت	حاصل عبارت	نوع عبارت
175 / 31	5	صحیح
175 % 31	20	صحیح
7.5 / 2	3.75	اعشاری double
7.5 % 2	1.5	اعشاری double
36 / 2.0	18.0	اعشاری double
36 % 2	0	صحیح
2541389 / 10	254138	صحیح
2541389 % 10	9	صحیح

عبارت های محاسباتی



$$23 / 5 = 4$$

$$23 \% 5 = 3$$



قواعد زیر را باید در هنگام انتساب یک عبارت به یک متغیر رعایت کنیم:

- ▶ اگر حاصل یک عبارت عدد صحیح باشد، بسته به بزرگی و اندازه عدد می تواند در یک متغیر نوع صحیح که گنجایش آن مساوی یا بزرگتر از حاصل عبارت باشد جای بگیرد.
- ▶ اگر حاصل یک عبارت از نوع صحیح باشد می تواند در یک متغیر نوع اعشاری نیز ذخیره شود، اما با این تفاوت که اعداد بزرگ فقط با 7 رقم دقت در نوع float و یا با 15 رقم دقت در نوع double ذخیره می شود و بقیه ارقام عدد گرد می شود.
- ▶ اگر حاصل یک عبارت از نوع اعشاری باشد نمی توان به طور ضمنی در یک متغیر نوع صحیح جای داده شود، فقط می تواند در یک متغیر اعشاری نوع float (باید بعد عدد F یا f نوشته شود) یا double جای گیرد.

نکته

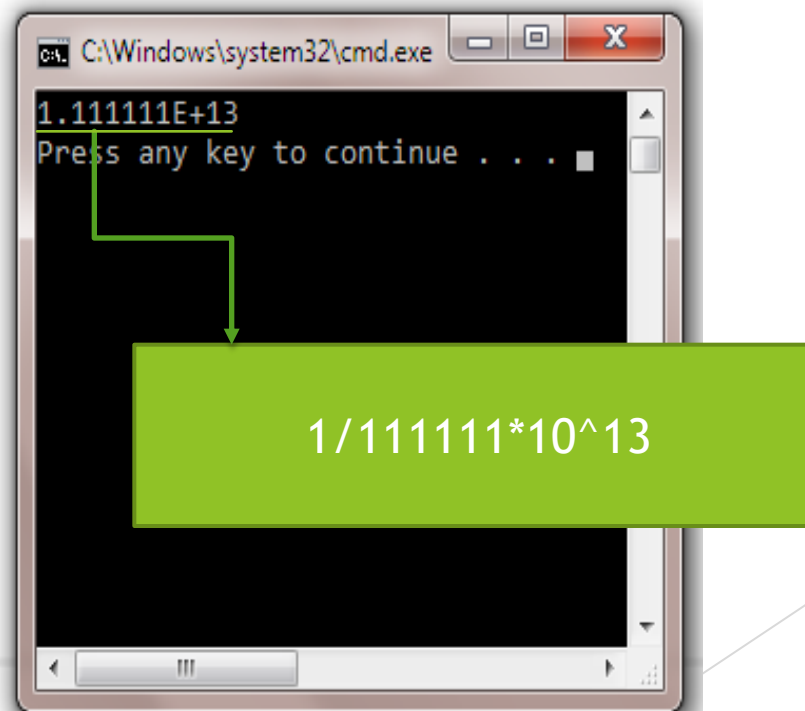
اگر در برنامه ای با عبارت ها و اعداد اعشاری با دقت حداکثر ۷ رقم سر و کار دارید و می خواهید از متغیرهای نوع float استفاده کنید، باید پس از هر عدد اعشاری، یک حرف f یا F قرار دهید تا مترجم آن عدد را به عنوان نوع float در نظر بگیرد. اما اگر از متغیرهای نوع double استفاده می کنید دیگر نیازی به نوشتن حرف f نیست.

```
float meanMark = 219.5f/14;
```

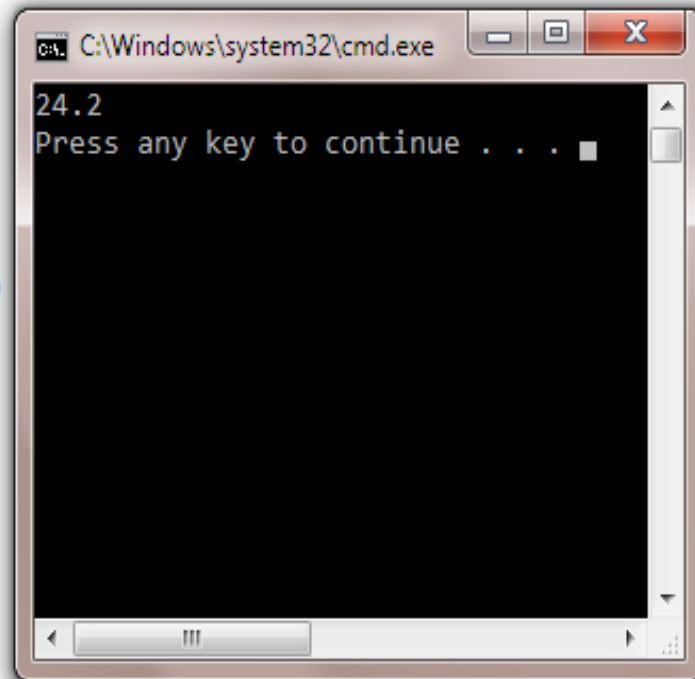
به دلیل اینکه حاصل عبارت به صورت صحیح است دیگر نیازی نیست F یا f نوشته شود. همچنین به دلیل اینکه float است تا هفت رقم دقت در نظر می گیرد.

```
namespace ConsoleApplication3
{
    class Program
    {
        static void Main(string[] args)
        {
            float a;
            a = 222222222222222/2;
            Console.WriteLine(a);
        }
    }
}
```

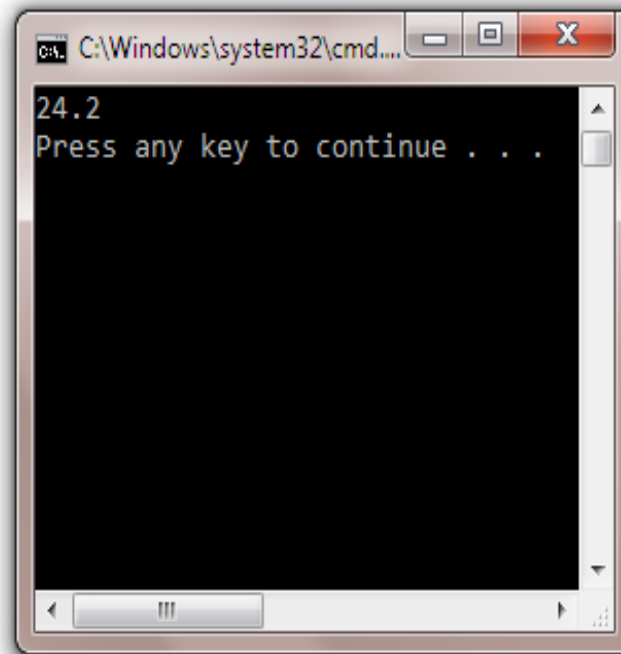
mohsen.rezghjoo@modares.ac.ir



```
namespace ConsoleApplication3
{
    class Program
    {
        static void Main(string[] args)
        {
            float a;
            a = 48.4f/2;
            Console.WriteLine(a);
        }
    }
}
```



```
namespace ConsoleApplication3
{
    class Program
    {
        static void Main(string[] args)
        {
            double a;
            a = 48.4/2;
            Console.WriteLine(a);
        }
    }
}
```



مثال: برنامه ای بنویسید که مجموع و معدل نمرات درسی جدول زیر را به دست آورد؟

► برای برنامه نویسی باید اول نمرات و تعداد واحد هر درس را در داخل متغیرهای مناسب قرار دهیم سپس تک تک نمره ها را در واحدهای خودشان ضرب کنیم و با هم جمع و در متغیر total mark قرار دهیم. سپس مجموع واحدها را جمع و در یک متغیر totalCredit قرار دهیم و در آخر با تقسیم مجموع نمرات بر مجموع واحدها معدل (average) را بدست آوریم.

تعداد واحد ^۱	نمره درس	نام درس
۳	۱۷/۵	فیزیک
۲	۱۹	شیمی
۴	۱۴/۵	ریاضی
۲	۱۸	ادبیات
۱	۱۹	ورزش ^۲

```
static void Main(string[] args)
{
    float physicMark = 17.5F, chemistryMark = 19, mathMark = 14.5F;
    float adabiyatMark = 18, varzeshMark = 19;
    int physicCredit = 3, chemistryCredit = 2, mathCredit = 4;
    int adabiyatCredit = 2, varzeshCredit = 1;
    float totalMark;
    int totalCredit;
    float average;

    totalMark=(physicMark*physicCredit)+
        (chemistryMark*chemistryCredit)+
        (mathMark*mathCredit)+
        (adabiyatMark*adabiyatCredit)+
        (varzeshMark*varzeshCredit);
```

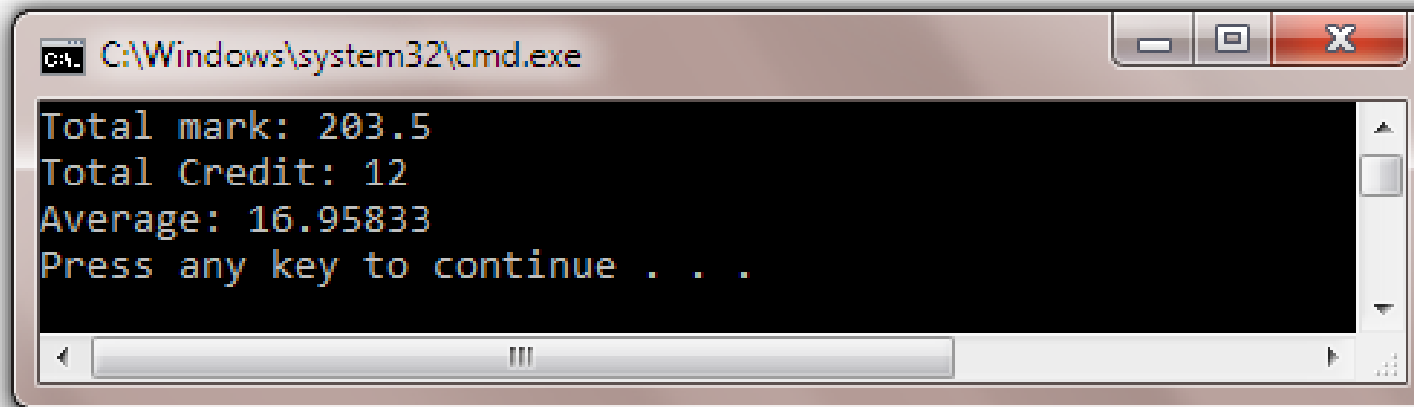


```
totalCredit = physicCredit + chemistryCredit +  
    mathCredit + adabiyatCredit + varzeshCredit;
```

```
average = totalMark / totalCredit;
```

```
Console.WriteLine("Total mark: "+totalMark);  
Console.WriteLine("Total Credit: "+totalCredit);  
Console.WriteLine("Average: "+average);
```

```
}
```

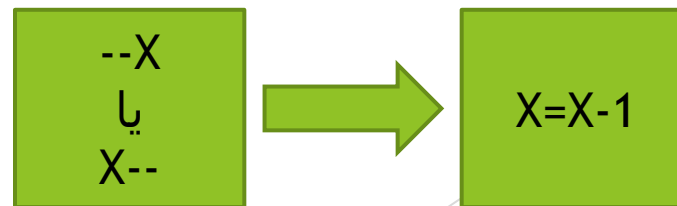
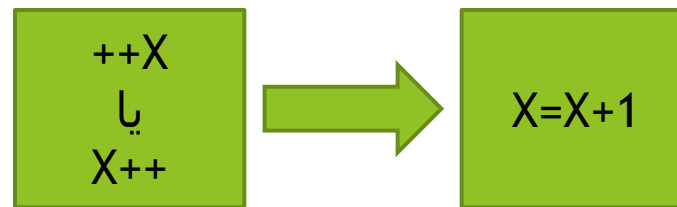


```
C:\Windows\system32\cmd.exe  
Total mark: 203.5  
Total Credit: 12  
Average: 16.95833  
Press any key to continue . . .
```


عملگرهای افزایشی (Increment) و کاهشی (Decrement)

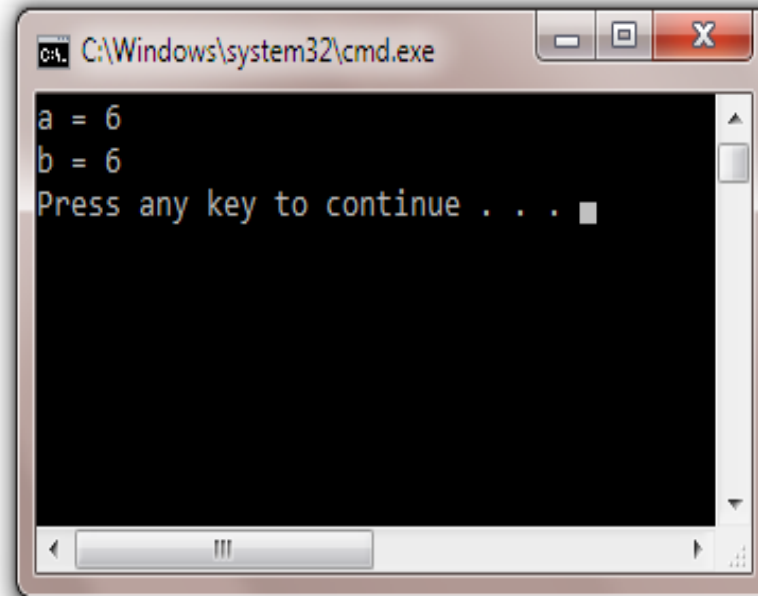
- ▶ در زبان سی شارپ عملگرهای یکتایی ++ و -- به ترتیب برای افزایش و کاهش مقدار یک متغیر به اندازه یک واحد در نظر گرفته می شوند.
- ▶ به عنوان مثال `int x=++a;` ابتدا مقدار متغیر a به مقدار یک واحد افزایش یافته و سپس در متغیر x ذخیره می شود.
- ▶ در `int x=a++;` ابتدا مقدار متغیر a در متغیر x ذخیره می شود و سپس مقدار a به یک واحد افزایش پیدا می کند.

مثال	نشانه	نوع عملگر	نام عملگر
<code>++value</code> یا <code>value++</code>	<code>++</code>	یکتایی	افزایش
<code>--value</code> یا <code>value--</code>	<code>--</code>	یکتایی	کاهش



عملگر افزایشی پیشوندی

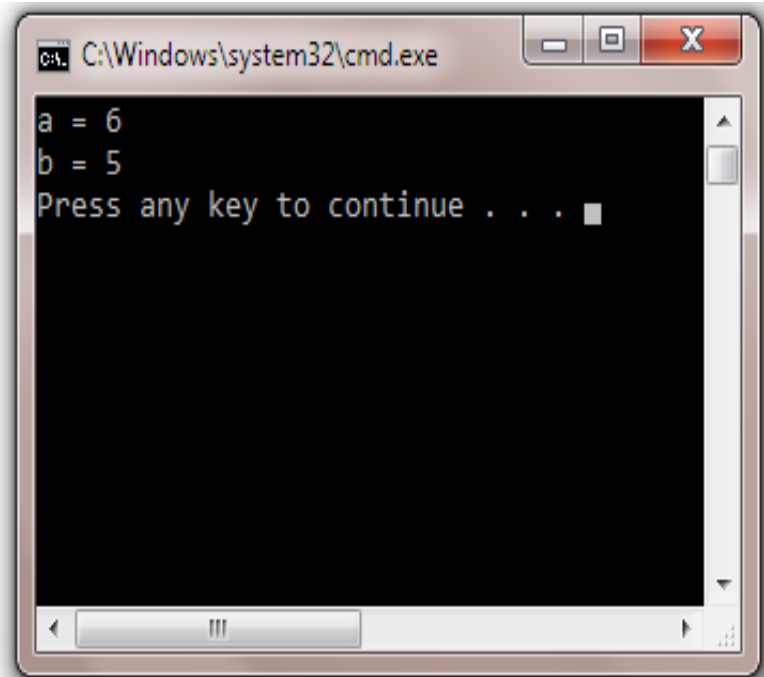
```
class Program
{
    static void Main(string[] args)
    {
        int a = 5;
        int b = ++a;
        Console.WriteLine("a = "+ a);
        Console.WriteLine("b = "+ b);
    }
}
```



```
C:\Windows\system32\cmd.exe
a = 6
b = 6
Press any key to continue . . .
```

عملگر افزایشی پسوندی

```
class Program
{
    static void Main(string[] args)
    {
        int a = 5;
        int b = a++;
        Console.WriteLine("a = "+ a);
        Console.WriteLine("b = "+ b);
    }
}
```

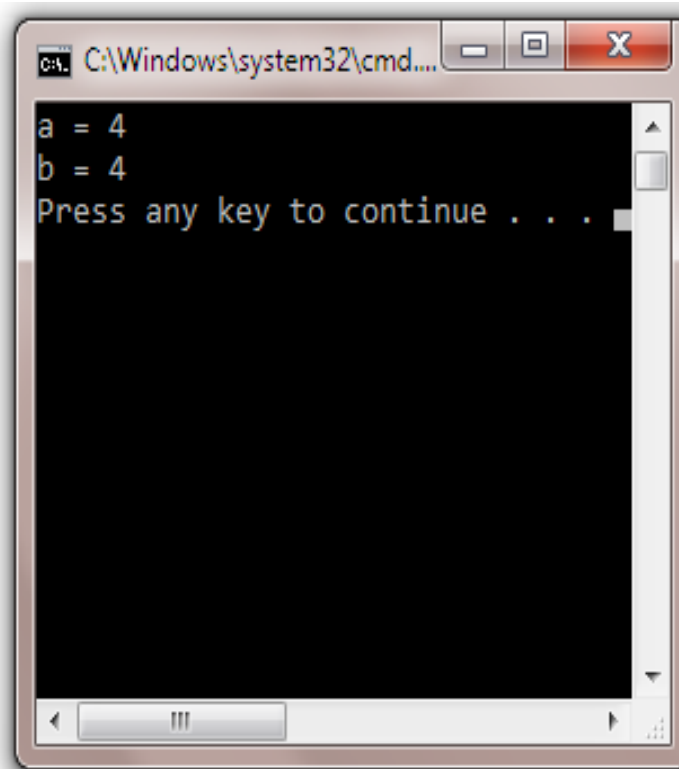


The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is displayed as follows:

```
a = 6
b = 5
Press any key to continue . . . █
```

عملگر کاهش‌دهنده پیشوندی

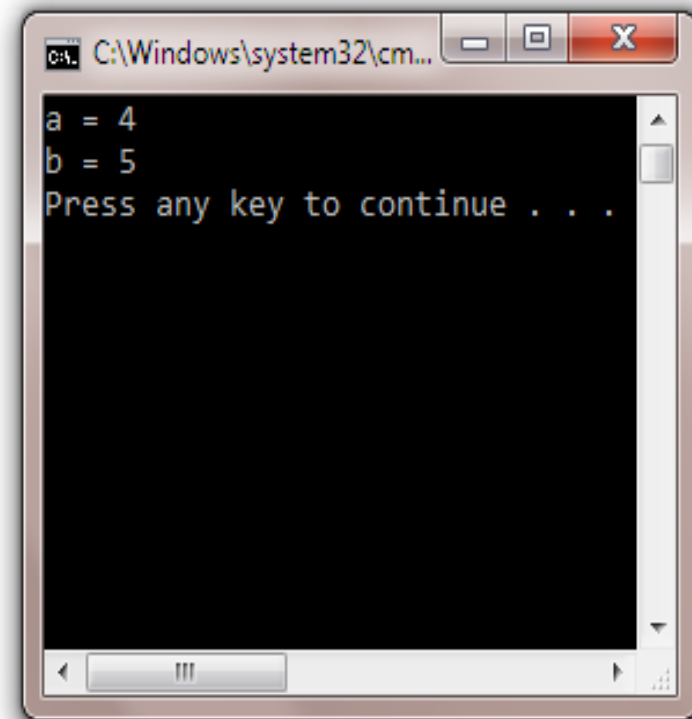
```
class Program
{
    static void Main(string[] args)
    {
        int a = 5;
        int b = --a;
        Console.WriteLine("a = " + a);
        Console.WriteLine("b = " + b);
    }
}
```



```
C:\Windows\system32\cmd...
a = 4
b = 4
Press any key to continue . . .
```

عملگر کاهش پسوندی

```
class Program
{
    static void Main(string[] args)
    {
        int a = 5;
        int b = a--;
        Console.WriteLine("a = "+ a);
        Console.WriteLine("b = "+ b);
    }
}
```



عملگرهای انتساب

- ▶ نباید بین علامت های این عملگرها فاصله گذاشت.
- ▶ به عنوان مثال $X+=4$ همان عبارت $X=X+4$ می باشد.
- ▶ به عنوان مثال $a*=2$ همان عبارت $a=a*2$ می باشد.
- ▶ به عنوان مثال $y/=a+b$ همان عبارت $y=y/(a+b)$ می باشد.

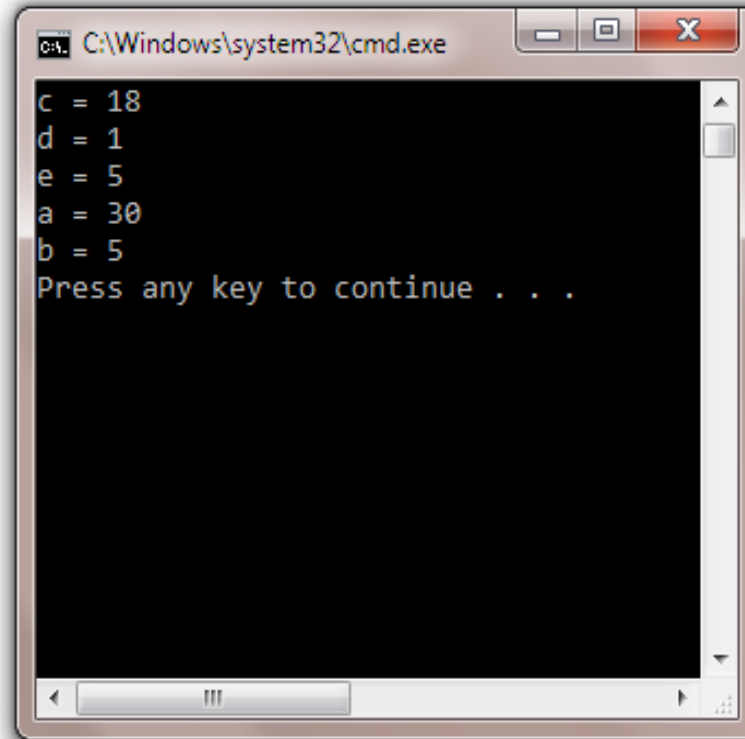
نشانه	نوع عملگر
$=$, $+=$, $--$, $*=$, $/=$, $\%=$ $\&=$, $ =$, $\wedge=$, $\ll=$, $\gg=$	انتساب

```
int z = y = x = 9;
```

عملگرهای انتساب

```
static void Main(string[] args)
{
    int a = 3, b=10, c=6, d=18, e=25;
    c *= a;
    d /= c;
    e %= b;
    a += 27;
    b = e;
    Console.WriteLine("c = "+ c);
    Console.WriteLine("d = "+ d);
    Console.WriteLine("e = "+ e);
    Console.WriteLine("a = "+ a);
    Console.WriteLine("b = " + b);
}
```

mohsen.rezghjoo@modares.ac.ir

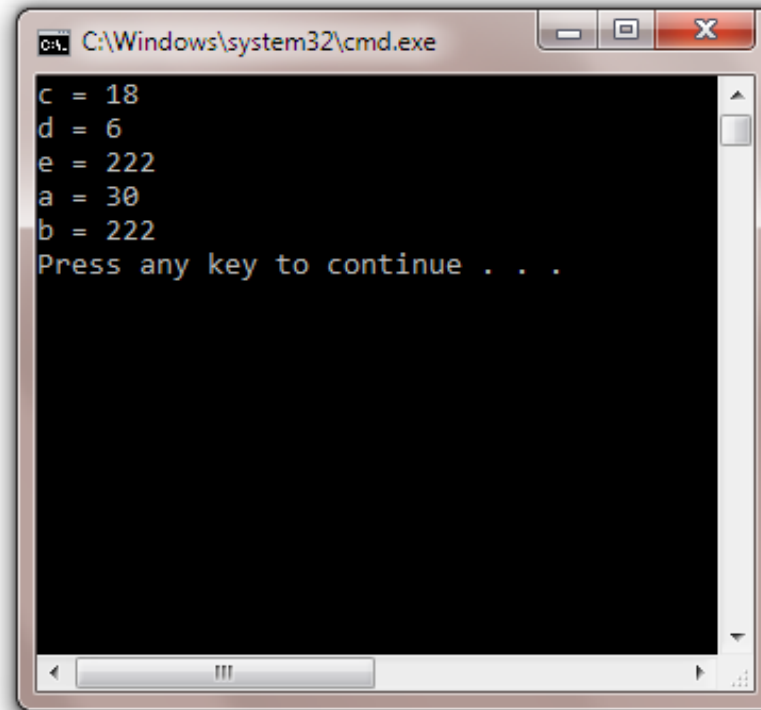


The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is displayed as follows:

```
c = 18
d = 1
e = 5
a = 30
b = 5
Press any key to continue . . .
```

عملگرهای انتساب

```
static void Main(string[] args)
{
    int a = 3, b=10, c=6, d=18, e=25;
    c *= a;
    d /= a;
    e %= b;
    a += 27;
    b = e = 222;
    Console.WriteLine("c = "+ c);
    Console.WriteLine("d = "+ d);
    Console.WriteLine("e = "+ e);
    Console.WriteLine("a = "+ a);
    Console.WriteLine("b = " + b);
}
```



```
C:\Windows\system32\cmd.exe
c = 18
d = 6
e = 222
a = 30
b = 222
Press any key to continue . . .
```


دلفوش از آنیم که حج میرویم
 غافل از آنیم که کج میرویم
 کعبه به دیدار خدا میرویم
 او که همین جاست کجا می رویم
 حج به خدا جز به دله پاک نیست
 شستن دل از دل غمناک نیست
 دین که به تسبیح و سر و ریش نیست
 هر که علی گفت که درویش نیست
 صبح به صبح در پی مکرو فریب
 شب همه شب ناله و امن یغیب



فصل ششم

دستورهای شرطی



دستورهای شرطی

▶ عبارت منطقی یا بولین :

- ▶ عبارت منطقی عبارتی است که حاصل آن فقط یکی از دو مقدار true یا false می باشد. به عنوان مثال عبارت $8 > 10$ یک عبارت منطقی است که نتیجه آن false است.
- ▶ معمولا در عبارت های منطقی از عملگرهای مقایسه ای استفاده می شود.

▶ عملگرهای مقایسه ای:

- ▶ از این عملگر برای مقایسه داده ها استفاده می شود و نتیجه آنها یکی از دو مقدار true یا false است. عملگرهای مقایسه ای می توانند دو عدد صحیح یا اعشاری و یا دو داده کاراکتر و یا رشته ای را با یکدیگر مقایسه کنند.
- ▶ برای بررسی مساوی یا برابر بودن دو مقدار از علامت == (دوبار علامت=) و برای بررسی مخالف یا نابرابر بودن دو مقدار، از علامت != استفاده می کنیم.

عملگرهای مقایسه ای

مثال	نشانه عملگر در ریاضی	عملگر در زبان C#	نام عملگر
<code>delta == 0</code>	=	==	برابری
<code>name != "AMIN"</code>	≠	!=	نامساوی
<code>max < number</code>	<	<	کوچک تر
<code>x <= a</code>	≤	<=	کوچک تر یا مساوی
<code>temperature > 25</code>	>	>	بزرگ تر
<code>(a+b) >= c</code>	≥	>=	بزرگ تر یا مساوی

عملگرهای مقایسه ای

```
static void Main(string[] args)
{
    int a = 50;
    Console.WriteLine(a == 50);
    Console.WriteLine(a >= 60);
    Console.WriteLine(a != 60);

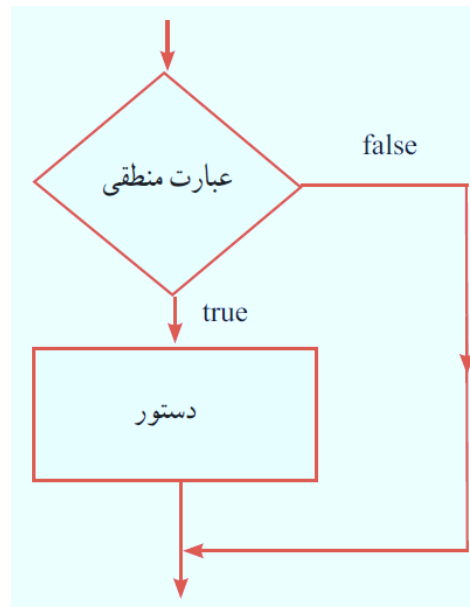
    char b='A';
    Console.WriteLine(b > 72);
    Console.WriteLine(b == 65);
    Console.WriteLine('B' == 'A'+1);
    Console.WriteLine('A' == 'B'+1);
}
```

A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window has a black background with white text. The output of the program is displayed as follows:
True
False
True
False
True
True
False
Press any key to continue . . .
The cursor is positioned at the end of the last line.

دستور شرطی if

- ▶ دستور شرطی if از سه بخش تشکیل شده است: 1- کلمه رزرو شده if (اگر) 2- عبارت منطقی داخل پارانتز، که در صورت صحیح بودن (true) عبارت منطقی دستور اجرا می شود 3-دستور

(عبارت منطقی) if
دستور ;



دستور شرطی if

نکته

به علامت نقطه ویرگول در دستور if توجه کنید. بعد از علامت پرانتز علامت نقطه ویرگول نگذارید، زیرا دستور if هنوز تمام نشده است. بلکه علامت نقطه ویرگول باید در انتهای دستور نوشته شود.
نقطه ویرگول ندارد

(عبارت منطقی) if

;دستور