

همان طور که در شکل ۲-۵ مشاهده می‌کنید مقداری که در متغیر number وجود دارد به صورت نماد علمی نشان داده می‌شود که اگر آن را به صورت معمولی تبدیل کنیم خواهیم داشت :

$$1.234568E\ 08 \rightarrow 1.234568 \times 10^8 \quad 123456800$$

اگر این مقدار را با حاصل عبارت اولیه مقایسه کنیم خواهیم دید که عدد با ۷ رقم گرد شده است :

$$123456789 \quad \text{حاصل عبارت}$$

$$123456800 \quad \text{مقدار ذخیره شده در متغیر}$$

۳- اگر حاصل یک عبارت از نوع اعشاری باشد **نمی‌تواند** به طور ضمنی، در یک متغیر نوع صحیح جای داده شود، فقط می‌تواند در یک متغیر اعشاری (نوع float و یا double) جای گیرد.

۴- اگر حاصل یک عبارت از نوع اعشاری double باشد، فقط در متغیر نوع double می‌تواند جای می‌گیرد. چرا؟

حاصل تقسیم یک عدد اعشاری بر یک عدد صحیح، عددی اعشاری است و مترجم آن را از نوع double در نظر می‌گیرد. تقسیم زیر را در نظر بگیرید :

$$219.5 / 14$$

با توجه به اینکه عدد 219.5 اعشاری است، بنابراین تقسیم نوع اعشاری انجام می‌شود و نتیجه عبارت از نوع double خواهد بود.

برای نگهداری نتیجه این عبارت، باید ابتدا متغیر مناسبی را تعریف کنیم و سپس با دستور انتساب حاصل عبارت را در آن ذخیره کنیم. با توجه به اینکه حاصل عبارت از نوع double است بنابراین باید متغیر نیز اعشاری و از نوع double باشد.

دستور زیر متغیری برای ذخیره معدل دانش آموز تعریف می‌کند تا نتیجه تقسیم را در آن ذخیره نماید.

```
Double meanScore;
```

اکنون می‌توانید مقدار عبارت را در متغیر مزبور با استفاده از دستور انتساب مقاردهی کنید.

```
Double meanScore = 219.5 / 14 ;
```

جایگزینی دو دستور بالا با یک دستور به صورت زیر خواهد بود :

```
Double meanScore = 219.5 / 14 ;
```

با اجرای دستور قبل عمل تقسیم انجام شده و حاصل تقسیم یعنی عدد 15.6786714285714 در متغیر meanScore ذخیره می‌شود.

**سؤال:** آیا لازم است معدل نمرات با چنین دقتی (۱۵ رقم) ذخیره شود؟

با توجه به اینکه معمولاً معدل نمرات با دو رقم اعشار بیان می‌شود، بنابراین در این موارد بهتر است از متغیر نوع float استفاده کنیم. در این صورت لازم است نوع عبارت محاسبه معدل نیز float باشد. برای اینکه نوع عبارت float شود می‌توانید از مترجم بخواهید عدد 219.5 را یک عدد نوع float در نظر بگیرد بدین منظور حرف f یا F را بعد از عدد می‌نویسیم 219.5 F  
بنابراین دستور زیر را می‌نویسیم:

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

با اجرای دستور بالا، عدد ۱۵/۶۷۸۵۷ در متغیر meanScore ذخیره می‌شود.  
اگر فراموش کنید که در دستور بالا، حرف f را بنویسید مترجم خطا می‌دهد. چون در سمت راست علامت انتساب، یک عبارت از نوع داده double است ولی در سمت چپ یک متغیر از نوع float است که ظرفیت کمتری نسبت به double دارد. در چنین حالتی در محیط VS یک خط قرمز زیر عبارت کشیده می‌شود و مترجم خطایی را صادر می‌کند (شکل ۳-۵).



شکل ۳-۵ خطای تبدیل نوع double به نوع float

شکل ۳-۵ بیان می‌کند که:

«مترجم نمی‌تواند نوع داده double را به طور ضمنی و خودکار به نوع float تبدیل کند.» باید به طور صریح و واضح از مترجم بخواهید عمل تبدیل نوع را انجام دهد.

## نکته

اگر در برنامه‌ای با عبارت‌ها و اعداد اعشاری با دقت حداکثر ۷ رقم سر و کار دارید و می‌خواهید 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 می‌کنید دیگر نیازی به نوشتن حرف f نیست.

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

مثال ۱-۵: مجموع و معدل نمرات درسی را طبق جدول ۳-۵ محاسبه کنید.  
جدول ۳-۵: نمرات درسی یک دانش آموز

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

به دلیل این که می‌خواهیم روی داده‌ها پردازش انجام دهیم بهتر است ابتدا نمرات را داخل متغیرهای مناسب ذخیره کنیم:

```
float physicMark 17.5F, chemistryMark 19, mathMark 14.5F;
```

```
float literacyMark 18, PEMark 19;
```

به همین صورت تعداد واحدها را نیز داخل متغیرهای مناسب قرار می‌دهیم:

```
int physicCredit 3, chemistryCredit 2, mathCredit 4;
```

```
int literacyCredit 2, PECredit 1;
```

حال می‌خواهیم مجموع نمرات را به دست آوریم. با توجه به اینکه تعداد واحد هر درس مختلف است نمی‌توانیم به سادگی نمرات را با یکدیگر جمع کنیم بلکه باید هر نمره را در تعداد واحد درسی مربوطه ضرب کنیم و سپس حاصل ضرب‌های به دست آمده را با یکدیگر جمع کنیم و نتیجه را در یک متغیر قرار دهیم. بنابراین متغیری به نام totalMark برای این منظور تعریف و حاصل عبارت محاسباتی مربوطه را در آن قرار می‌دهیم:

```
float totalMark ;
```

```
totalMark (physicMark * physicCredit)
```

```
(chemistryMark * chemistryCredit)
```

```
(mathMark * mathCredit)
```

```
(literacyMark * literacyCredit)
```

```
(PEMark * PECredit);
```

**سؤال:** این عبارت محاسباتی نیازی به پرانتز گذاری ندارد. چرا؟ در اینجا فقط برای بالابردن خوانایی عبارت یا برنامه از پرانتزها استفاده کرده‌ایم.

پس از به دست آوردن مجموع نمرات که در متغیر totalMark، قرار گرفته است، مجموع واحدها را نیز حساب می‌کنیم و در یک متغیر به نام totalCredit قرار می‌دهیم.

```
int totalCredit = physicCredit + chemistryCredit + mathCredit + literacyCredit + PECredit;
```

در این لحظه می‌توانیم با تقسیم مجموع نمرات بر مجموع واحدها، معدل را به دست آوریم و در یک متغیر ذخیره کنیم.

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

برای نمایش مقادیر متغیرها از دستورات زیر استفاده می‌کنیم:

```
Console.WriteLine("Total mark: " + totalMark);
```

```
Console.WriteLine("Total credit: " + totalCredit);
```

```
Console.WriteLine("Average: " + average);
```

با توجه به توضیحات بالا برنامه مورد نظر چنین خواهد بود:

```
using System;
```

```
class Average
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        float physicMark = 17.5F, chemistryMark = 19, mathMark = 14.5F;
```

```
        float literacyMark = 18, PEMark = 19;
```

```
        int physicCredit = 3, chemistryCredit = 2, mathCredit = 4;
```

```
        int literacyCredit = 2, PECredit = 1;
```

```
        float totalMark;
```

```
        totalMark = (physicMark * physicCredit)
```

```
                    (chemistryMark * chemistryCredit)
```

```
                    (mathMark * mathCredit)
```

```
                    (literacyMark * literacyCredit)
```

```
                    (PEMark * PECredit);
```

```

int totalCredit = physicCredit + chemistryCredit + mathCredit + literacyCredit
+ PECredit;

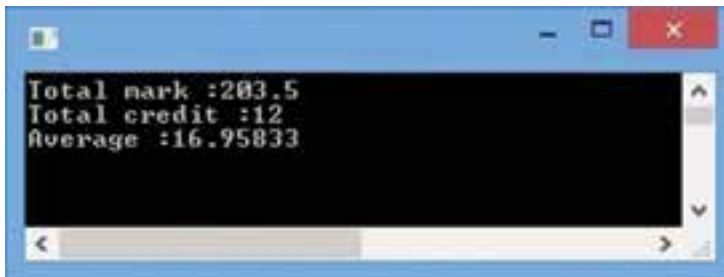
float average = totalMark / totalCredit;

Console.WriteLine("Total mark: " + totalMark);
Console.WriteLine("Total credit: " + totalCredit);
Console.WriteLine("Average: " + average);
Console.ReadKey();
}
}

```

برنامه ۵-۱- محاسبه معدل نمرات یک دانش آموز

خروجی برنامه چنین است :



شکل ۵-۴- خروجی برنامه ۵-۱

به مثال ۵-۱ دستوراتی اضافه کنید که نمرات درسی را از ورودی دریافت کند. این برنامه را در محیط VS بنویسید و آن را ذخیره، ترجمه و اجرا نمایید.

### ۳-۵- عملگرهای افزایشی<sup>۱</sup> و کاهشی<sup>۲</sup>

در زبان برنامه نویسی C#، عملگرهای یکتایی و -- به ترتیب برای افزایش و کاهش مقدار یک متغیر به اندازه یک واحد در نظر گرفته شده است. البته این عملگرها از زبان C وارد این زبان شده اند و برای کوتاه شدن و تایپ کمتر دستورات، ابداع شده اند که در زبان‌های دیگر امروزی مانند جاوا نیز وجود دارند (جدول ۴-۵).

جدول ۴-۵- (عملگرهای افزایشی و کاهشی)

نام عملگر	نوع عملگر	نشانه	مثال
افزایش	یکتایی		value یا value
کاهش	یکتایی		value یا value

اکنون می‌توانیم اضافه کردن یک نمره به درس ریاضی را با استفاده از عملگر افزایشی انجام

دهیم:

```
++mathMark; // عملگر افزایشی قبل از نام متغیر قرار گرفته است
```

یا

```
mathMark++; // عملگر افزایشی بعد از نام متغیر قرار گرفته است
```

عملگر افزایشی یا کاهشی را می‌توانید قبل از نام متغیر و یا بعد از نام متغیر ذکر کنید که در هر دو حالت سبب می‌شود مقدار متغیر به اندازه یک واحد تغییر کند. تفاوت بین این دو حالت را در سؤالات ۱۱ و ۱۲ خودآزمایی ببینید.

<sup>۱</sup> Increment

<sup>۲</sup> Decrement

## ۴-۵ عملگرهای انتساب

در جدول ۵-۵ عملگرهای انتساب دیده می‌شود.

نوع عملگر	نشانه
انتساب	, * , / , %
	& , ^ , << , >>

با عملگر و کاربرد آن در فصل قبل آشنا شدید که برای ذخیره کردن یک مقدار در یک متغیر استفاده می‌شود. دستورات زیر را در نظر بگیرید.

```
int x , y ;  
x 6;  
string helloString "Hello World";  
y x;
```

در دستور آخر، مقداری که در متغیر x قرار دارد (عدد ۶) به متغیر y منتسب می‌شود و y نیز برابر ۶ می‌شود.

### نکته

حاصل عبارتی که دارای عملگر انتساب است، مقدار داده یا متغیری است که در سمت راست عملگر واقع شده است.

$x \leftarrow 6$  یعنی  $x$  6;

مثلاً در دستور زیر

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

چند عملگر انتساب وجود دارد و از سمت راست به ترتیب انجام می‌شوند. ابتدا عدد ۹ در متغیر x قرار می‌گیرد و سپس حاصل عبارت که عدد ۹ است در متغیر y کپی شده و سپس این مقدار در متغیر z کپی می‌شود. یعنی در عبارتی که عملگرهای انتساب وجود دارد این عملگرها از سمت راست به چپ انجام می‌شوند (شرکت پذیر راست) بر خلاف عملگرهای ریاضی مشابه که از چپ به راست انجام می‌شوند.

  
int z y x 9;

اکنون اگر بخواهیم یک نمره، به نمره درس ریاضی در مثال ۱-۵ اضافه کنیم، با توجه به اینکه نمره درس را در یک متغیر ذخیره کردیم، دستور لازم برای افزایش یک واحد به متغیر مربوطه را بنویسید.

```
mathMark 1
```

متغیر mathMark حاوی نمره درس ریاضی است. برای افزایش یک نمره از دستور انتساب

زیر استفاده می‌کنیم:

```
mathMark mathMark 1 ;
```

با اجرای این دستور انتساب، ابتدا کامپیوتر عبارت سمت راست علامت را محاسبه می‌کند. بدین منظور محتوای متغیر mathMark که برابر با ۱۴/۵ است با عدد یک جمع می‌شود و حاصل عبارت یعنی ۱۵/۵ را در متغیر سمت چپ، جایگزین مقدار قبلی آن می‌کند. بنابراین از این به بعد محتوای متغیر مزبور ۱۵/۵ خواهد بود که به این ترتیب یک واحد افزایش یافته است.

با دستوری مشابه، می‌توانیم مقدار یک متغیر را کاهش دهیم مثلاً اگر بخواهید دو واحد از مجموع تعداد واحدها کم کنید دستور زیر را می‌نویسیم:

```
totalCredit totalCredit 2 ;
```

در این دستور نیز ابتدا عبارت سمت راست محاسبه و نتیجه در متغیر سمت چپ قرار می‌گیرد. در جدول ۵-۵، علاوه بر عملگر ، عملگرهای دیگر انتساب همراه با یک عملگر ریاضی وجود دارند. عملگرهای / و \* هر یک، علاوه بر عمل انتساب، یک عمل ریاضی را نیز روی یک عملوند مشترک انجام می‌دهند تا تایپ دستورات خلاصه شود.

فرض کنید بخواهید محتوای متغیر x را سه برابر کنید در این صورت از دستور زیر استفاده می‌کنید:

```
x *= 3;
```

**سؤال:** معادل دستور  $x *= 3$  چه دستوری است؟

**مثال ۲-۵:** با اجرای دستورات زیر چه عددی بر روی خروجی نشان داده می‌شود؟

```
int x = 3;
```

```
int y = 4;
```

```
x *= y;
```

```
Console.WriteLine(x);
```

در دستور  $x * y$  حاصل ضرب متغیر x در متغیر y محاسبه می‌شود (برابر ۱۲) و در متغیر x قرار می‌گیرد. بنابراین عدد ۱۲ در روی صفحه نمایش داده می‌شود.

**سؤال:** معادل دستور  $x * y$  چه دستوری است؟



۱- عبارت، عملگر و عملوند را با ذکر یک مثال مشخص کنید.

۲- عبارت ریاضی  $4x + \frac{5x}{y+3}$  را با توجه به جدول ۱-۵ (اولویت عملگرها) به صورت یک عبارت قابل قبول در C#، بنویسید.

۳- مترجم در ترجمه کدام یک از دستورات زیر، خطا می‌دهد؟ دلیل خود را ذکر کنید.

1) `int number 127.9;`

2) `float number 12 2.3;`

3) `ushort myMark 1300;`

4) `sbyte totalScore 250;`

۴- اگر در متغیر `number` یک عدد صحیح قرار داشته باشد، دستوری بنویسید که رقم یکان آن را بدست آورده و در خروجی نشان دهد.

۵- در دستورات زیر، نمودار اجرای عملگرها را مانند شکل ۱-۵ در ابتدای این فصل، رسم کنید و حاصل را به دست آورید.

`int n 600 10 * 9 % 4 ;`

`System.Console.WriteLine( 12 * 2 % 8 3 * 2 );`

۶- برای انجام عبارت‌های ستون سمت راست، از چه عملگری در ستون چپ استفاده می‌کنید؟

الف - `%` ۱- کاهش ۶ واحد از یک متغیر

ب - ۲- افزایش یک واحد به یک متغیر

ج - ۳- کاهش یک واحد از یک متغیر

د - ۴- افزایش `n` واحد به یک متغیر

ه - ۵- محاسبه رقم یکان یک عدد صحیح

۷- خروجی قطعه کد زیر را به دست آورید. اعداد نشان داده شده در خروجی، مانند چه دنباله‌ای از اعداد در ریاضیات است؟

```
n 0;
d 4;
a 5;
Console.WriteLine(a d * n);
n 1;
Console.WriteLine(a d * n);
n 1;
Console.WriteLine(a d * n);
n 1;
Console.WriteLine(a d * n);
```

در متدهای `WriteLine()` بالا، با چه فرضی فضای نامی `System` در ابتدای هر خط، ذکر نشده است؟

۸- اگر متغیر `Z` را یک بار با عدد ۵ و بار دیگر با عدد ۳ مقداردهی کنیم،

دستور `Console.WriteLine(z)` چه مقداری را در نهایت چاپ می‌کند؟ توضیح دهید؟

۹- برنامه‌ای بنویسید که دو عدد از ورودی دریافت نماید و مقادیر درون دو متغیر را جابجا کند.

(از مثال جا بجایی محتوای دو لیوان حاوی شیر و نوشابه کمک بگیرید)

۱۰- دستورات زیر بر روی مقادیر متغیرهای `a, b` چه عملی را انجام می‌دهد؟

پاسخ را در جدول `Trace` وارد کنید.

```
int a 5;
int b 10;
a a b;
b a b;
a a b;
```

a	b

۱۱- نتیجه اجرای دستورات زیر چیست؟

```
int a 5;  
int b a;  
Console.WriteLine(a);  
Console.WriteLine(b);
```

۱۲- نتیجه اجرای دستورات زیر چیست؟ نتیجه این تمرین را با تمرین قبلی مقایسه کنید.

```
int a 5;  
int b a ;  
Console.WriteLine(a);  
Console.WriteLine(b);
```

۱۳- سؤال زیر به زبان انگلیسی است آن را خوانده و پاسخ‌های صحیح را مشخص کنید.

Which of the following is the correct way to initialize the variables i and j to a value 10 each?

A) `int i 10;`

B) `int i 10,j 10;`

`int j 10;`

C) `int i,j;`

D) `int i,j 10;`

`i 10: j 10;`

E) `int i j 10;`

## تمرینات برنامه‌نویسی فصل پنجم

۱- نیروی جاذبه در کره ماه  $\frac{1}{6}$  نیروی جاذبه در زمین است. برنامه‌ای بنویسید که وزن یک شخص در روی زمین را سؤال کرده و سپس وزن وی در کره ماه را حساب کند.

۲- اگر محیط<sup>۱</sup> یک مربع<sup>۲</sup> برابر ۱۸ سانتی متر باشد برنامه‌ای بنویسید که اطلاعات زیر را روی صفحه نشان دهد:

الف) اندازه هر ضلع<sup>۳</sup> مربع بر حسب سانتی متر

ب) اندازه مساحت<sup>۴</sup> مربع بر حسب سانتی متر مربع

۳- برنامه‌ای بنویسید که محیط یک مربع را سؤال کند و سپس اندازه ضلع و اندازه مساحت مربع را محاسبه نماید.

۴- سال ۱۴۰۰ شمسی را در نظر بگیرید. تعداد روزهای سپری شده تا این سال چند روز است؟ تعداد ساعت‌های سپری شده چقدر است؟ برنامه‌ای بنویسید که شماره سال را از کاربر سؤال کند و سپس تعداد روز و ساعت سپری شده را در متغیرهای مناسبی قرار دهد و سپس نمایش دهد.

۵- یک فروشگاه پوشاک، اجناس خود را با ۱۵ درصد تخفیف، حراج کرده است، برنامه‌ای بنویسید که مبلغ قبل از تخفیف جنس را از ورودی دریافت کند و سپس قیمت بعد از تخفیف را محاسبه و نمایش دهد.

۶- برای یک دستگاه خودپرداز برنامه‌ای بنویسید که یک عدد را به عنوان مبلغ یک اسکناس دریافت کند و آن را به اسکناس‌های کوچکتر خرد<sup>۵</sup> کند. مثلاً بتواند یک اسکناس ۱۰۰ هزار ریالی را به دو اسکناس ۵۰۰۰۰ ریالی و یا ۱۰ اسکناس ۱۰۰۰۰ ریالی خرد کند.  
(راهنمایی: از عملگرهای % و / استفاده کنید.)



۱\_ Per meter

۲\_ Square

۳\_ S de

۴\_ Area

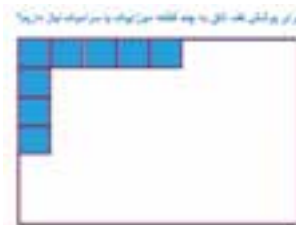
۵\_ Change

- ۷- برنامه‌ای بنویسید که مقلوب یک عدد صحیح دو رقمی دریافتی از کاربر را نمایش دهد.
- ۸- برنامه‌ای بنویسید که تاریخ تولد هر فرد را بر حسب ماه، روز و سال دریافت نماید و سپس تعداد روزهای عمر وی را حساب کند. هر سال ۳۶۵ روز و هر ماه ۳۰ روز است.
- ۹- (این سؤال نیاز به برنامه‌نویسی ندارد و فقط یک محاسبه ساده است.)



می‌خواهیم کف یک اتاق  $4 \times 3$  متری را سرامیک کنیم. اگر هر قطعه سرامیک به شکل مربع و به اندازه  $5^\circ$  سانتی متر باشد به چند قطعه سرامیک نیاز داریم؟ اگر اندازه آن برابر  $4^\circ$  سانتی متر باشد آن وقت به چه تعداد سرامیک نیاز داریم؟

۱۰- برنامه‌ای به نام Simple Tiling Calculator بنویسید که اندازه طول (Width) و عرض (Length) یک اتاق را بپرسد و سپس اندازه یک سرامیک مربع شکل را سؤال نماید. تعداد سرامیک‌های لازم برای پوشش کف را محاسبه و تعیین کند. توجه داشته باشید که نتیجه به دست آمده از محاسبات باید عدد صحیح باشد. مثلاً عددی مانند  $17/25$  باید به عدد ۱۸ گرد شود.



۱۱- مدیر یک ساختمان مسکونی قصد دارد، هزینه برق عمومی ساختمان، را بر اساس تعداد نفرات ساکنین تقسیم کند. تعداد نفرات هر خانواده طبق جدول زیر است، برنامه‌ای بنویسید تا به مدیر ساختمان در بدست آوردن هزینه برق هر خانواده کمک کند. برنامه باید یک عدد به عنوان هزینه برق از کاربر دریافت کند و سپس با استفاده از اطلاعات جدول، هزینه برق هر خانوار را محاسبه و نمایش دهد.

شماره واحد	تعداد نفرات خانواده
۱	۳
۲	۴
۳	۲
۴	۵
۵	۲

## دستورهای شرطی

در بیشتر برنامه‌های کاربردی، لازم است مقدار داده‌ها را بررسی و مقایسه کنیم. سپس بر اساس نتیجه حاصل از بررسی، دستور یا دستورهایی را اجرا کنیم. به عبارت دیگر، در برنامه‌ها لازم است بتوانیم بر اساس مقدار داده‌ها، تصمیم‌گیری کنیم. برنامه‌هایی که تاکنون نوشته‌ایم تمام دستورهای داخل متد (Main)، پشت سرهم و به نوبت اجرا می‌شدند. اکنون در این فصل با دستورهای شرطی آشنا می‌شویم که به وسیله آنها اجرای دستورها و پردازش آنها، کنترل می‌شوند و رفتار برنامه بر اساس وضعیت داده‌ها تغییر می‌کند.

### پس از پایان این فصل انتظار می‌رود که فراگیر بتواند:

- ۱- انواع عملگرهای مقایسه‌ای را توضیح دهد و آنها را در برنامه استفاده کند.
- ۲- ساختار دستور if-else را در برنامه‌های شرطی خود استفاده کند.
- ۳- مفهوم عملگرهای منطقی را شرح دهد و انواع آنها را نام ببرد.
- ۴- از عملگرهای منطقی به طور صحیح در برنامه‌ها استفاده نماید.
- ۵- با استفاده از جدول عملگرهای منطقی، نتیجه عبارتهای ترکیبی منطقی و مقایسه‌ای را به دست آورد.
- ۶- اولویت عملگرها را با استفاده از جدول تقدم عملگرها در حل عبارات ترکیبی به کاربندد.
- ۷- دستور switch را در برنامه‌های خود به کارگیرد.
- ۸- با ترکیب دستورهای شرطی، برنامه‌هایی با ساختار پیچیده شرطی بنویسد.

## ۱-۶- عبارت منطقی یا بولین<sup>۱</sup>

در فصل چهارم در قسمت معرفی انواع داده‌ها با نوع داده منطقی، آشنا شدید. این نوع داده، فقط دارای دو مقدار true (درست) یا false (نادرست) است. عبارت منطقی نیز عبارتی است که حاصل آن فقط یکی از دو مقدار true یا false است.

مثلاً عبارت  $10 > 12$  یک عبارت منطقی است، که نتیجه آن false است چون 12 کوچکتر از 10 نیست. ولی در عبارت  $x > 0$  به شرط اینکه x عددی مثبت باشد نتیجه عبارت true است.

در دو مثال بالا از عملگرهای مقایسه‌ای < یا > استفاده شده است، عملگرهای مقایسه‌ای دیگری نیز وجود دارند که در جدول ۱-۶، مشاهده می‌شود. این عملگرها شبیه عملگرهایی است که در ریاضیات استفاده می‌شود. معمولاً در عبارات منطقی از عملگرهای مقایسه‌ای استفاده می‌شود.

## ۲-۶- عملگرهای مقایسه‌ای<sup>۲</sup>

همان طور که از نام این عملگرها مشخص است، برای مقایسه داده‌ها استفاده می‌شوند و نتیجه آنها یکی از دو مقدار true یا false است. عملگرهای مقایسه‌ای می‌توانند دو عدد صحیح یا اعشاری و یا دو داده کاراکتری و یا رشته‌ای را با یکدیگر مقایسه کنند.

علامت بعضی از عملگرهای مقایسه‌ای در زبان C#، با علامت ریاضی آنها کمی متفاوت است مثلاً برای بررسی مساوی یا برابر بودن دو مقدار، از علامت `==` استفاده می‌شود (دو بار علامت =) و یا برای بررسی مخالف یا نابرابر بودن دو مقدار، از علامت `!=` استفاده می‌شود.

در جدول ۱-۶ هر یک از عملگرهای مقایسه‌ای به همراه علامت ریاضی آنها نشان داده شده است.

---

۱- Boolean Expression

۲- Comparison Operators

جدول ۱-۶- عملگرهای مقایسه‌ای

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

مثال ۱-۶: در برنامه ۱-۶ حاصل چند عبارت منطقی بر روی صفحه نشان داده می‌شود:

using System;

class Expression

```
{
    static void Main()
    {
        int weight = 700;
        Console.WriteLine(weight > 500); // True
        char gender = 'm';
        Console.WriteLine(gender == 'f'); // False
        double colorWaveLength = 1.630;
        Console.WriteLine(colorWaveLength > 1.621); // True
        Console.WriteLine('B' < 'A'); // True
        Console.ReadKey();
    }
}
```

برنامه ۱-۶- عبارت‌های منطقی



حاصل عبارت‌های منطقی را می‌توانید بر روی صفحه خروجی نمایش دهید و یا در داخل متغیرهایی از نوع bool ذخیره کنید.

### ۳-۶- دستور شرطی if

در درس مبانی کامپیوتر، با دستور شرطی «اگر» آشنا شدید و به وسیله آن می‌توانستید اجرای دستورها را کنترل کنید. در زبان برنامه‌نویسی C# از دستور if (با حروف کوچک نوشته می‌شود) برای کنترل اجرای دستورها و بررسی شرط، استفاده می‌شود. ساختار کلی دستور if به صورت زیر است:

(عبارت منطقی) if

; دستور

دستور شرطی if از سه بخش تشکیل شده است: کلمه رزرو شده if، عبارت منطقی داخل پرانتز و دستوری که در صورت درست (true) بودن نتیجه عبارت، اجرا خواهد شد. توجه داشته باشید که پس از کلمه if، یک جفت پرانتز وجود دارد و عبارتی از نوع منطقی که برای بررسی و مقایسه داده است، داخل پرانتز نوشته می‌شود. در خط بعدی، دستوری که می‌خواهیم در صورت درست بودن عبارت منطقی اجرا شود، با تورفتگی می‌نویسیم و در انتهای آن علامت ; را به منزله پایان دستور if قرار می‌دهیم. در دستورات زیر نمونه‌ای از به کارگیری دستور if را می‌بینید.

```
if (mark < 10)
```

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

**سؤال؟** اگر در متغیر mark، نمره ۸/۵ قرار داشته باشد، به نظر شما با اجرای دستورهای بالا، چه پیامی بر روی صفحه نمایش، ظاهر می‌شود؟ برای نمره ۱۸ چطور خواهد بود؟

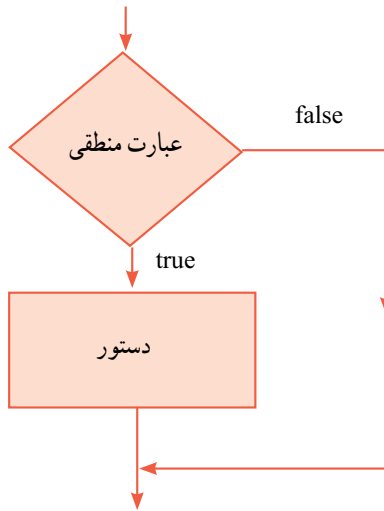
به علامت نقطه ویرگول در دستور if توجه کنید. بعد از علامت پرانتز علامت نقطه ویرگول نگذارید، زیرا دستور if هنوز تمام نشده است. بلکه علامت نقطه ویرگول باید در انتهای دستور نوشته شود.

نقطه ویرگول ندارد

(عبارت منطقی) if

دستور;

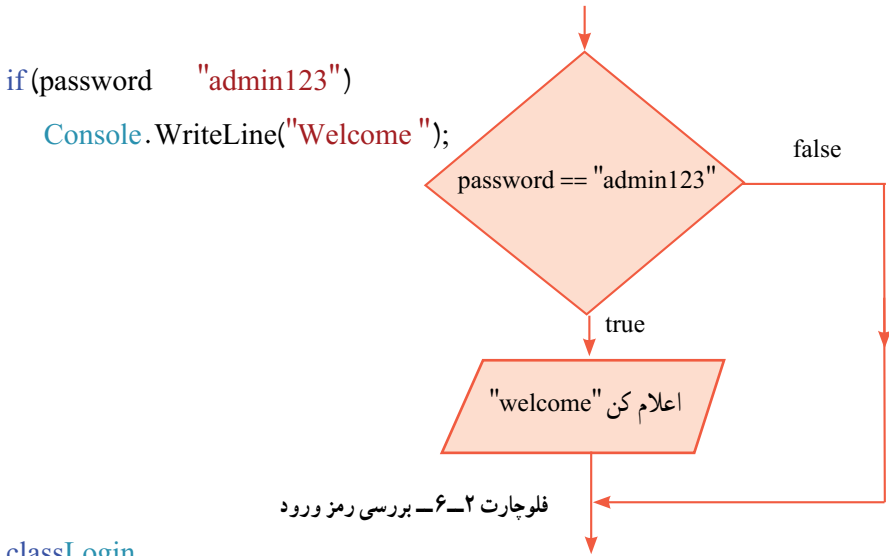
فلوجارت دستور if ساده به صورت زیر است :



فلوجارت ۱-۶- دستور if

وقتی که کامپیوتر، در حال اجرای برنامه است با رسیدن به دستور if، ابتدا مقدار عبارت را محاسبه می‌کند. در صورتی که ارزش عبارت، true باشد، دستوری که بعد از if قرار دارد، اجرا می‌شود و در صورتی که مقدار عبارت false باشد، دستور مربوطه اجرا نمی‌شود.

مثال ۲-۶: می‌خواهیم برنامه‌ای بنویسیم که رمز ورود را از کاربر دریافت کند و در صورت صحیح بودن پیام مناسب چاپ نماید(رمز صحیح admin123).  
 الگوریتم یا روش انجام کار: ابتدا رمز را از کاربر دریافت کرده و در متغیر password قرار می‌دهیم. اگر رمز وارد شده مساوی admin123 بود پیام Welcome را نمایش می‌دهیم. دستور شرطی این برنامه در زیر آورده شده است.



```

classLogin
{
    static void Main()
    {
        string password;
        Console.Write("Enter password: ");
        password = Console.ReadLine();
        if (password == "admin123")
            Console.WriteLine("Welcome");
        Console.WriteLine("Press any key to continue...");
        Console.ReadKey();
    }
}
    
```

برنامه ۲-۶- برنامه بررسی رمز ورود