

عبارت‌های محاسباتی

یکی از توانایی‌های کامپیوتر قدرت و سرعت بالا در انجام محاسبات است. انجام بیش از یک میلیارد عمل جمع بر روی اعداد نوع صحیح در کمتر از یک ثانیه، بسیار شگفت آور است. کامپیوترها با استفاده از چنین توانایی قادر هستند عملیات و پردازش مورد نظر در برنامه را با سرعت بر روی داده‌ها انجام دهند. از جمله پردازش‌هایی که از کامپیوتر استفاده می‌شود، جستجوی داده مورد نظر در بین داده‌ها یا عمل مرتب‌سازی داده‌ها به ترتیب خاصی است. برای انجام چنین پردازش‌هایی لازم است ابتدا با انواع عبارت‌ها و محاسبات در زبان C# آشنا شویم و سپس از آنها در برنامه‌های خود استفاده کنیم.

پس از پایان این فصل انتظار می‌رود که فراگیر بتواند:

- ۱- عملوند، عملگر و عبارت محاسباتی را تعریف کند و آنها را به درستی به کارگیرد.
- ۲- عملگرهای ریاضی را نام ببرد و در عبارت محاسباتی از آنها استفاده کند.
- ۳- حاصل عبارت محاسباتی را با استفاده از جدول تقدم عملگرها به دست آورد.
- ۴- کاربرد عملگرهای افزایشی، کاهش‌ی و انتساب را بیان نماید.

۱-۵- عبارت چیست؟

در درس ریاضی با عبارت‌های محاسباتی مختلفی مانند عبارت‌های زیر آشنا شدید.

$$8 \quad 3 * 5$$

$$9 - 7.25$$

$$5.6 * 3 \quad 5.6$$

$$\frac{y}{6} \quad 25 \quad x$$

۱- Express on

در این عبارات‌ها، علامت نشانه عمل جمع و علامت * نشانه عمل ضرب است. به این علامت‌ها که بیانگر انجام یک عمل بر روی اعداد و داده‌ها هستند، عملگر^۱ گفته می‌شود. مثلاً عملگر * در عبارت $5 * 3$ بر روی اعداد 3 و 5 عمل ضرب را انجام می‌دهد و همچنین عملگر در عبارت بالا بر روی عدد 8 و نتیجه حاصل ضرب یعنی عدد 15، عمل جمع را انجام می‌دهد. به اعدادی که یک عملگر بر روی آنها عملی را انجام می‌دهد عملوند^۲ می‌گویند. اعداد 3 و 5 عملوندهای عملگر ضرب و عدد 8 و 15 عملوندهای عملگر جمع هستند.

هر یک از عملگرهای ضرب و جمع بر روی دو عدد عمل می‌کنند و به عبارتی دارای دو عملوند هستند به این عملگرها، عملگرهای دوتایی^۳ گفته می‌شود. عملگر در عبارت $7.25 - 9$ عملگر تفریق است که آن نیز یک عملگر دوتایی است و حاصل تفریق 7.25 از 9 را محاسبه می‌کند. اما عملگر قرینه در عبارت x ، فقط دارای یک عملوند x است و آن را قرینه می‌کند. این عملگر یک عملگر یکتایی^۴ است.

نکته

یک عبارت از تعدادی عملگر و عملوند تشکیل شده است و دارای یک حاصل یا نتیجه می‌باشد. نتیجه یا حاصل یک عبارت ممکن است عددی یا غیر عددی باشد.

۲-۵- عملگرهای ریاضی یا حسابی^۵

اگر در عبارتی بیش از یک عملگر وجود داشته باشد ابتدا عملگری عمل خود را انجام می‌دهد که اولویت^۶ بالاتری نسبت به دیگری داشته باشد. مثلاً اولویت عملگر ضرب بیش از اولویت عملگر جمع است. چنانچه دو یا چند عملگر دوتایی، با اولویت یکسان در یک عبارت وجود داشته باشد ابتدا عملگر سمت چپ انجام می‌شود. به عبارت دیگر از سمت چپ به راست، عملگرها به ترتیب انجام می‌شوند که به آن «شرکت پذیری چپ^۷» می‌گویند.

در جدول ۱-۵ لیست عملگرهای ریاضی را به ترتیب اولویت مشاهده می‌کنید. عملگر قرینه اولویت بالاتری نسبت به بقیه عملگرهای ریاضی دارد و عملگرهای جمع و تفریق دارای اولویت یکسان ولی کمترین اولویت را در بین عملگرهای ریاضی دارند.

۱- Operator

۲- Operand

۳- Binary Operator

۴- Unary Operator

۵- Arithmetic

۶- Precedence

۷- Left-Associative

جدول ۱-۵- عملگرهای ریاضی

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	

عملکرد عملگرهای جمع و تفریق و ضرب مانند عملکرد آنها در ریاضیات است اما عملگر تقسیم با توجه به نوع عملوندهایش می تواند تقسیم صحیح و بدون ممیز و یا تقسیم اعشاری و ممیزی انجام دهد. مثلاً در عبارت $9/2$ چون عملوندها اعداد صحیح هستند بنابراین تقسیم بدون ممیز و صحیح انجام خواهد شد که نتیجه آن عدد ۴ است. اما در عبارت $9.0/2$ یا در عبارت $9/2.0$ ، چون حداقل یکی از عملوندها، اعشاری است بنابراین تقسیم به صورت اعشاری انجام می شود که حاصل عبارت عدد $4/5$ است.

در جدول ۱-۵، عملگر جدیدی نیز به نام باقیمانده تقسیم که با نشانه % مشخص می شود، مشاهده می کنید. به وسیله این عملگر می توانیم باقیمانده تقسیم یک عدد بر عدد دیگر را با توجه به خارج قسمت صحیح و بدون اعشار به دست آوریم. مثلاً در تقسیم عدد ۲۳ بر عدد ۵، خارج قسمت بدون اعشار عدد ۴ است بنابراین باقیمانده عدد ۳ است.

$$23 / 5 = 4$$

$$23 \% 5 = 3$$

۲۳		۵
۲۰		۴
—		۳

برای تغییر دادن اولویت عملگرها، از علامت های پرانتز استفاده می شود. مثلاً در عبارت زیر ابتدا عمل جمع و سپس عمل ضرب انجام می شود.

$$5.6 * (3 + 6.5)$$

اگر چند پرانتز تو در تو نیز وجود داشته باشد، ابتدا داخلی ترین پرانتز انجام می شود.

در جدول ۵-۲، چند نمونه از عبارات‌های ریاضی نشان داده شده است.

جدول ۵-۲- مثالی از عبارات‌های ریاضی

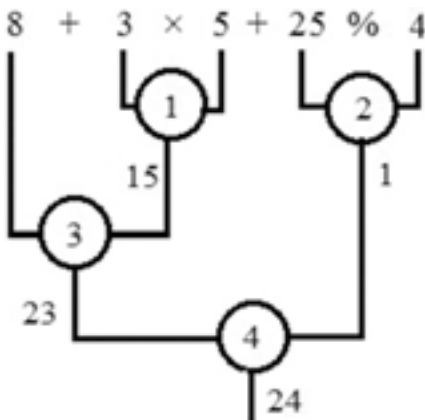
عبارت	حاصل عبارت	نوع عبارت
$175 / 31$	5	صحیح
$175 \% 31$	20	صحیح
$7\ 5 / 2$	3 75	اعشاری double
$7\ 5 \% 2$	1 5	اعشاری double
$36 / 2\ 0$	18 0	اعشاری double
$36 \% 0\ 2$	0	صحیح
$2541389 / 10$	254138	صحیح
$2541389 \% 10$	9	صحیح

عبارات محاسباتی زیر را در نظر می‌گیریم و سپس نوع آن را تعیین می‌کنیم.

$$8\ 3 * 5\ 25 \% 4$$

در این عبارت بیش از یک عملگر وجود دارد، بنابراین ابتدا عملگری که دارای اولویت بالاتر است، انجام می‌شود. چون اولویت عملگرهای * و % بالاتر از عملگر + است بنابراین ابتدا این دو

عملگر انجام می‌شود و از طرفی چون این دو عملگر دارای اولویت یکسان هستند، برطبق شرکت‌پذیری چپ، ابتدا عملگر سمت چپ یعنی * و سپس عملگر % انجام می‌شود. بر همین اساس در مورد عملگرهای جمع نیز ابتدا عملگر سمت چپ و سپس عملگر بعدی انجام می‌شود (نمودار ۵-۱).



نمودار ۵-۱- ترتیب اجرای عملگرها

در برنامه‌ها، معمولاً حاصل یا نتیجه یک عبارت را در یک متغیر نگهداری می‌کنند. البته نوع متغیری که حاصل یک عبارت، در آن قرار می‌گیرد باید با نوع عبارت، سازگار باشد. مانند ظرفی در آشپزخانه که بخواهیم در آن غذا یا نوشیدنی بریزیم باید گنجایش مناسب آن غذا را داشته باشد. قوانین زیر، باید به وسیله برنامه نویس، در هنگام انتساب یک عبارت به یک متغیر، رعایت شود، در غیر اینصورت با پیام خطای مترجم مواجه می‌شویم. مترجم زبان C# روی این قوانین سخت‌گیر است زیرا می‌خواهد از اشتباهات برنامه‌نویسان جلوگیری نماید، این یکی از ویژگی‌های زبان C# است.

۱- اگر حاصل یک عبارت عدد صحیح باشد بسته به اندازه و بزرگی عدد، می‌تواند در یک متغیر نوع صحیح که گنجایش آن مساوی یا بزرگ‌تر از حاصل عبارت باشد جای گیرد. مثلاً حاصل عبارت 31 / 175 عدد 5 است این عدد کوچک می‌تواند در تمام متغیرهای نوع صحیح زیر قرار گیرد.

`sbyte , byte , short , ushort , int , uint , long , ulong`

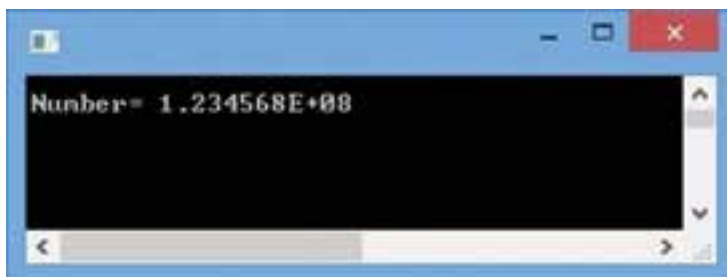
ولی در عبارت 10 / 2541389 چون حاصل عبارت عدد 254138 است که عدد صحیح بزرگی است فقط در متغیرهای نوع `int , uint , long , ulong` قابل نگهداری است.

۲- اگر حاصل یک عبارت از نوع صحیح باشد می‌تواند در یک متغیر نوع اعشاری نیز ذخیره شود، اما با این تفاوت که اعداد بزرگ (`long`) فقط با ۷ رقم دقت (در نوع `float`) و یا با ۱۵ رقم دقت (در نوع `double`) ذخیره می‌شود و بقیه ارقام عدد، گرد می‌شود.

مثلاً در دستور زیر حاصل عبارت 10 / 1234567890 در متغیر اعشاری ذخیره می‌شود ولی به دلیل اینکه عدد بزرگی است به صورت گرد شده در متغیر ذخیره می‌شود (شکل ۲-۵).

`float number = 1234567890 / 10;`

`Console.WriteLine("Number " + number);`



شکل ۲-۵- نتیجه اجرای عملگر تقسیم

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.

$$\text{Double meanScore};$$

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

$$\text{Double meanScore} \quad 219.5 / 14 ;$$

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

$$\text{Double meanScore} \quad 219.5 / 14 ;$$

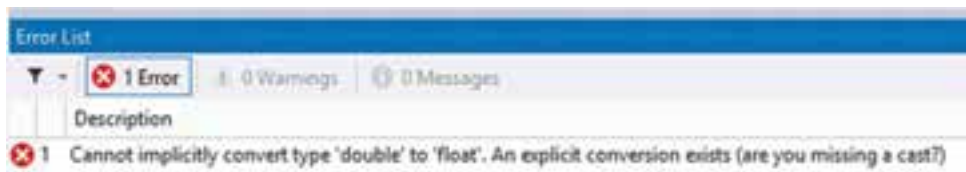
با اجرای دستور قبل عمل تقسیم انجام شده و حاصل تقسیم یعنی عدد 15.6786714285714 در متغیر meanScore ذخیره می‌شود.

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

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;

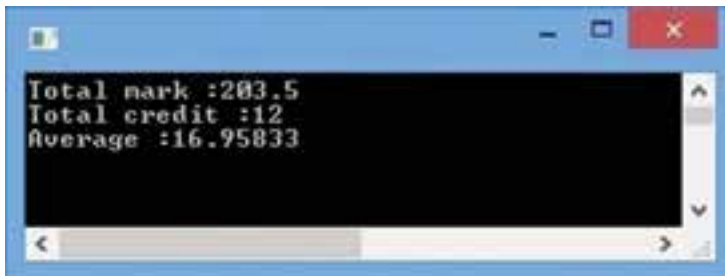
float average = totalMark / totalCredit;

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 :



```

Total mark :283.5
Total credit :12
Average :16.95833

```

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^۱ و کاهششی^۲

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 کاهششی)

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 کاهششی 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.

^۱ Increment

^۲ 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 6; یعنی $x \leftarrow 6$

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 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 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^۱ یک مربع^۲ برابر ۱۸ سانتی متر باشد برنامه‌ای بنویسید که اطلاعات زیر را روی صفحه نشان دهد:

الف) اندازه هر ضلع^۳ مربع بر حسب سانتی متر

ب) اندازه مساحت^۴ 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^۵ 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

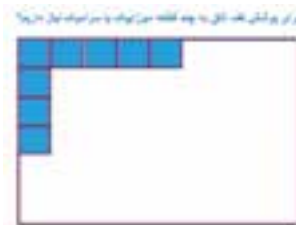
۵_ Change

- 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 4×3 متری را سرامیک کنیم. اگر هر قطعه سرامیک به شکل مربع و به اندازه 5° سانتی متر باشد به چند قطعه سرامیک نیاز داریم؟ اگر اندازه آن برابر 4° 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^۱

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^۲

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 کارا کتری 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' + 1); // 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؟

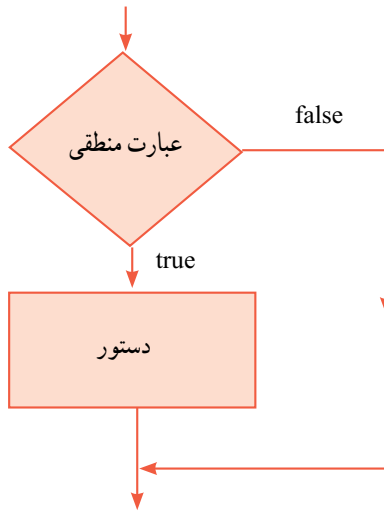
به علامت نقطه ویرگول در دستور if توجه کنید. بعد از علامت پرانتز علامت نقطه ویرگول نگذارید، زیرا دستور if هنوز تمام نشده است. بلکه علامت نقطه ویرگول باید در انتهای دستور نوشته شود.

نقطه ویرگول ندارد

(عبارت منطقی) if

دستور;

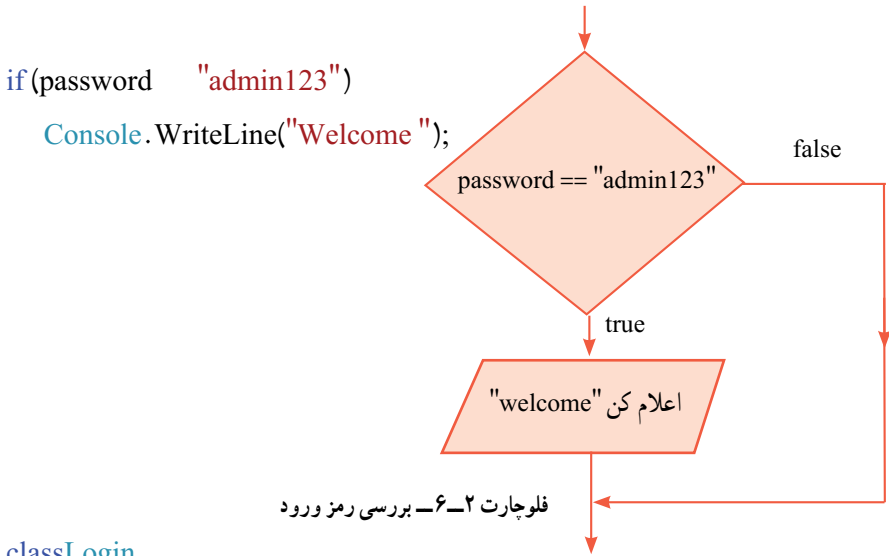
فلوجارت دستور if ساده به صورت زیر است :



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.



```

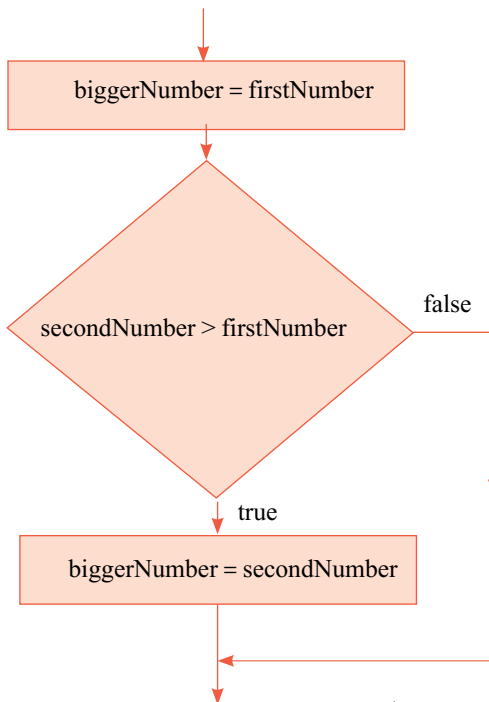
class Login
{
    static void Main()
    {
        string password;
        Console.Write("Enter password: ");
        password = Console.ReadLine();
        if (password == "admin123")
            Console.WriteLine("Welcome");
        Console.WriteLine("Press any key to continue...");
        Console.ReadKey();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

برنامه ۲-۶- برنامه بررسی رمز ورود

۱- برنامه ۲-۶ را در محیط VS بنویسید، ترجمه و اجرا نمایید و به ازای ورودی‌های مختلف برنامه را آزمایش کنید.

۲- مثال ۳-۶: می‌خواهیم برنامه‌ای بنویسیم، که عدد بزرگتر را از بین دو عدد دریافتی کاربر، پیدا کند.

الگوریتم یا روش انجام کار: ابتدا اعداد مورد نظر را از کاربر دریافت کرده و در دو متغیر `firstNumber` و `secondNumber` قرار می‌دهیم. برای پیدا کردن بزرگ‌ترین عدد، ابتدا اولین عدد را به عنوان عدد بزرگتر فرض می‌کنیم و داخل یک متغیر مثلاً به نام `biggerNumber` قرار می‌دهیم. آن‌گاه مقدار این متغیر (فرض خود را) را با عدد بعدی مقایسه می‌کنیم. اگر این عدد همچنان بزرگتر بود نیازی به انجام کاری نیست اما اگر عدد دوم، بزرگتر بود، لازم است فرض اولیه خود را تصحیح کنیم و عدد دوم را به عنوان عدد بزرگتر در نظر بگیریم. به این ترتیب در متغیر `biggerNumber` عدد دوم را قرار می‌دهیم. در انتها نیز، مقدار موجود در `biggerNumber` را نمایش می‌دهیم. اگر بیش از دو عدد در اختیار داشتیم و بخواهیم عدد بزرگتر را پیدا کنیم، باز هم از همین روش یا الگوریتم می‌توانیم استفاده کنیم. فلوجارت ۳-۶ را مشاهده کنید.



فلوجارت ۳-۶- پیدا کردن عدد بزرگتر از بین دو عدد

```

class FindMaximum
{
    static void Main()
    {
        string input;
        int firstNumber , secondNumber ;

        Console.WriteLine("Enter a number: ");
        input = Console.ReadLine();
        firstNumber = int.Parse(input);

        Console.WriteLine("Enter another number: ");
        input = Console.ReadLine();
        secondNumber = int.Parse(input);

        int biggerNumber = firstNumber;
        if (secondNumber > firstNumber)
            biggerNumber = secondNumber;

        Console.WriteLine("The maximum number is: " + biggerNumber);

        Console.WriteLine("Press any key to continue...");
        Console.ReadKey();
    }
}

```

دریافت عدد اول

دریافت عدد دوم

برنامه ۳-۶ برنامه تشخیص عدد بزرگتر از بین دو عدد

توجه داشته باشید که در نوشتن دستورهای بالا، همه دستورها به جز خط بعد از دستور if، زیر هم نوشته می شوند.

سؤال؟ به نظر شما چرا دستور `biggerNumber = secondNumber;` از ابتدای خط شروع نشده است و به اندازه یک tab به سمت داخل نوشته شده است؟

دلیل این کار، این است که می‌خواهیم نشان دهیم که این دستور، یک دستور عادی نیست بلکه اجرای این دستور، وابسته به ارزش عبارت منطقی دستور if است. بنابراین همواره سعی کنید دستور مربوط به دستور if را کمی داخل تر بنویسید تا برنامه خوانا و واضح باشد. اگر چه رعایت نکردن این قاعده، تاثیری در ترجمه و اجرا ندارد و اگر حتی تمام دستورها را پشت سر هم و در یک خط طولانی نیز بنویسید باز برنامه، ترجمه می‌شود اما اشکال زدایی و یا گسترش برنامه بسیار سخت خواهد بود و درک برنامه نیز مشکل می‌شود.

مثال ۴-۶: عددی صحیح، داخل متغیری به نام number قرار دارد و می‌خواهیم زوج بودن آن را پس از تشخیص، با پیامی مناسب اعلام کنیم. الگوریتم یا روش انجام کار: می‌دانیم عددی زوج است که رقم یکان آن یکی از اعداد ۰ و ۲ و ۴ و ۶ و ۸ باشد. از طرفی می‌دانیم که عدد زوج بر ۲ بخش پذیر است یعنی باقیمانده تقسیم آن بر عدد ۲ صفر است. از یکی از این دو روش می‌توانیم به سادگی زوج بودن عدد را تشخیص دهیم. در این مثال، از روش بخش پذیری بر ۲ استفاده می‌کنیم. ابتدا باقیمانده تقسیم عدد بر ۲ را حساب می‌کنیم، اگر باقیمانده برابر صفر بود پیامی را روی صفحه نمایش می‌دهیم تا نشان دهد که عدد زوج است.

```
int remainder = number % 2;
```

```
if (remainder == 0)
```

```
    Console.WriteLine(number + " is Even.");
```

اگر در متغیر number، عددی زوج مانند ۱۶ باشد، حاصل باقیمانده تقسیم آن عدد بر ۲، صفر خواهد شد و در این صورت نتیجه عبارت منطقی در دستور if، مقدار true خواهد بود. بنابراین دستور نمایش پیام اجرا می‌شود و اعلام می‌کند که عدد زوج است. اما اگر در دستوره‌های بالا، عددی که در متغیر number قرار دارد، عددی فرد مانند ۱۵ باشد، حاصل باقیمانده تقسیم آن عدد بر ۲ عدد یک خواهد شد و در این صورت نتیجه عبارت منطقی در دستور if، مقدار false خواهد بود. بنابراین دستور نمایش پیام، اجرا نمی‌شود و چیزی روی صفحه نشان داده نمی‌شود.

یکی از اشتباهات رایج برنامه نویسان در نوشتن عبارت منطقی دستور `if`، استفاده از علامت `=` به جای `==` است. در این صورت، مترجم متوجه بروز چنین اشتباهی می‌شود و علاوه بر کشیدن خط قرمز در زیر عبارت مورد نظر، خطایی را صادر می‌کند (شکل ۶-۱).

```

int remainder = number % 2;
if (remainder = 0)
    Console.WriteLine(number + " is Even");

Console.ReadKey();
}
}
}

```

00 % - <

Error List

1 Error | 0 Warnings | 0 Messages

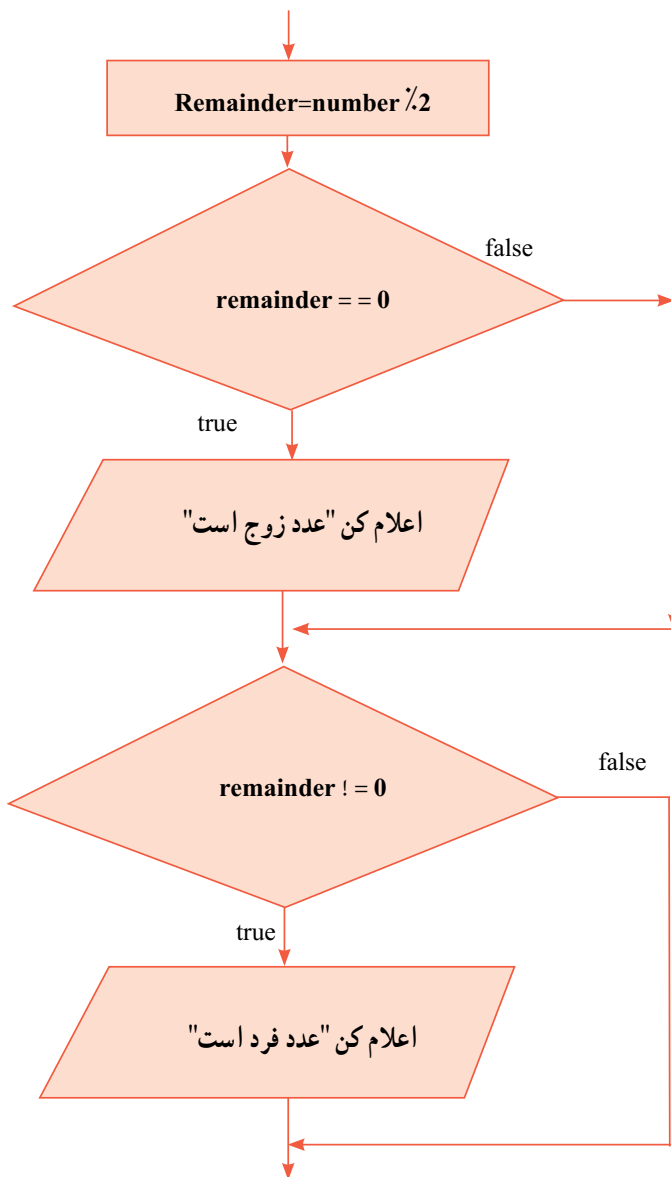
Description

1 Cannot implicitly convert type 'int' to 'bool'

شکل ۶-۱- اشتباه رایج برنامه نویسان و بروز خطای مترجم

مثال ۶-۵: اکنون می‌خواهیم دستورهای مربوط به مثال ۶-۴ را طوری توسعه دهیم که اگر در متغیر `number`، عددی فرد وجود داشت، آن را شناسایی کرده، و پیام مناسب اعلام کند.

الگوریتم یا روش انجام کار: برای انجام این کار می‌توانیم یک دستور `if` به دستورهای قبلی اضافه کنیم که برای تشخیص عدد فرد، استفاده شود. اگر باقیمانده تقسیم عدد بر ۲، برابر یک شود آن عدد فرد است. بنابراین پیام "عدد فرد است" نمایش داده می‌شود. از آنجا که حاصل باقیمانده هر عددی بر ۲ عدد صفر یا یک است، بنابراین به جای شرط مساوی یک، از شرط نامساوی صفر استفاده می‌کنیم. فلوجارت ۶-۴ را مشاهده کنید.



فلوچارت ۴-۶- تشخیص زوج و فرد با دو دستور if

مطابق با فلوجارت ۴-۶، دستورهای لازم برای انجام کار، چنین خواهد بود :

```
int remainder = number % 2;
if (remainder == 0)
    Console.WriteLine(number + "is Even");
else
    Console.WriteLine(number + "is Odd");
```

۴-۶- دستور شرطی if-else

در مثال ۴-۶، برای اینکه عدد زوج یا فرد را تشخیص دهیم از باقیمانده تقسیم عدد بر ۲ استفاده کردیم. در فلوجارت شکل ۴-۶، تصمیم گیری بر اساس باقیمانده تقسیم صورت می گیرد که عدد صفر یا مخالف است. در چنین مواردی از دستور شرطی if-else می توان استفاده کرد. شکل کلی این دستور چنین است :

```
if (عبارت منطقی)
    دستور شماره ۱
else
    دستور شماره ۲
```

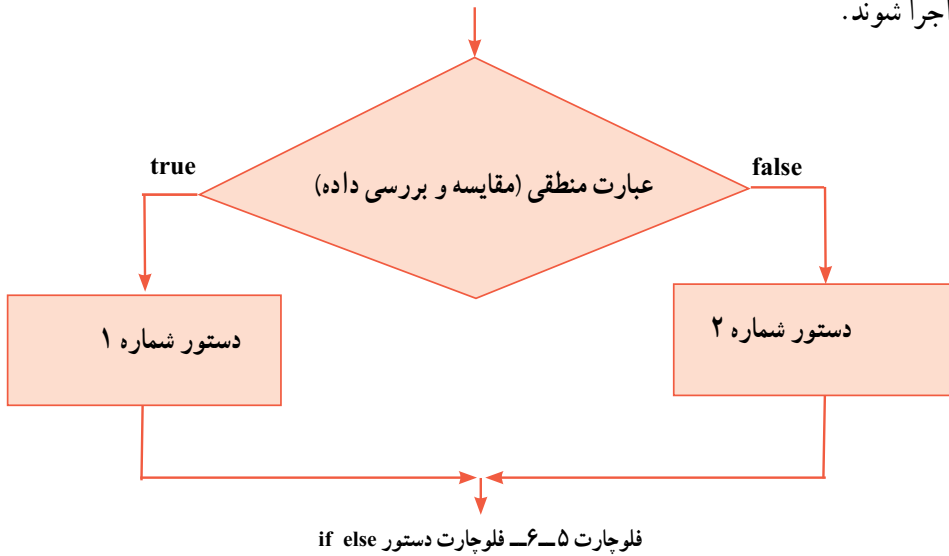
اگر نتیجه عبارت منطقی true باشد، دستور شماره ۱ که مربوط به قسمت if است اجرا می شود و در غیر این صورت، یعنی اگر نتیجه عبارت منطقی false باشد، دستور شماره ۲ که مربوط به قسمت else است، اجرا می شود.

نکته

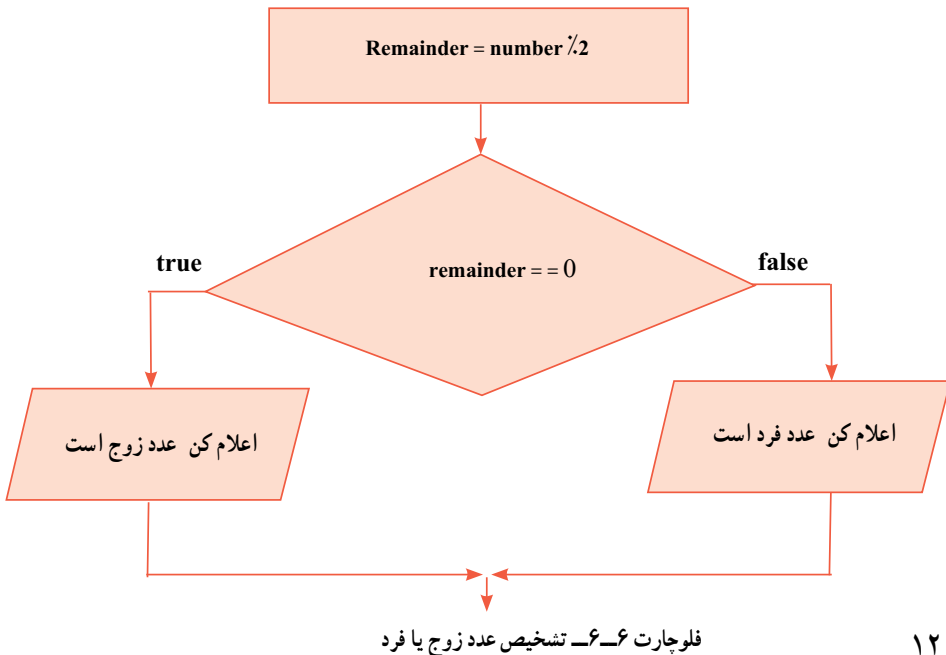
دستور if else را خوانا بنویسید یعنی :

دستورهای شماره ۱ و ۲ که مربوط به قسمت if یا قسمت else است را با کمی تورفتگی بنویسید تا معلوم شود که هر کدام متعلق به چه قسمتی است. مراقب باشید که بعد از کلمه رزرو شده else، علامت نقطه ویرگول قرار ندهید.

دستور if-else به صورت فلوجارت ۵-۶ در زیر نشان داده شده است. توجه داشته باشید که همواره فقط یکی از دو دستور شماره ۱ یا شماره ۲ اجرا می‌شود و محال است که هر دو دستور با هم اجرا شوند.



مثال ۶-۶: در این مثال، دستورهای تشخیص زوج یا فرد بودن عدد را با استفاده از دستور if-else، طبق فلوجارت ۶-۶ می‌نویسیم.



```
int remainder = number % 2;
if (remainder == 0)
    Console.WriteLine(number + " is Even");
else
    Console.WriteLine(number + " is Odd");
```

روش‌های که در این مثال استفاده شد، نسبت به روشی که در مثال ۴-۶ به کار گرفته شد را با یکدیگر مقایسه کنید. کدام یک بهتر است؟

کار در کارگاه ۲

مثال ۷-۶: می‌خواهیم برنامه‌ای بنویسیم که حقوق دریافتی (خالص) یک کارمند را محاسبه نماید. هر کارمند دارای یک حقوق ثابت است که با توجه به میزان تحصیلات و تجربه کاری معین می‌شود. از حقوق تمام کارمندان، ۷ درصد به عنوان حق بیمه کسر می‌گردد. همچنین کارکنانی که حقوق آنها بیش از ده میلیون ریال باشد، مالیات به اندازه ۵ درصد کسر می‌گردد (مالیات به مازاد بر ده میلیون تعلق می‌گیرد).

الگوریتم یا روش انجام کار: فلوجارت ۷-۶ عملیات محاسبه حقوق را نشان می‌دهد. با توجه به فلوجارت، ابتدا حقوق ثابت کارمند دریافت می‌شود. حق بیمه آن محاسبه و سپس حقوق با عدد ده میلیون ریال مقایسه می‌شود. چنانچه حقوق بیش از ده میلیون ریال باشد، آن‌گاه مقدار مالیات محاسبه و بر روی صفحه نشان داده می‌شود. در غیر این صورت مالیات صفر در نظر گرفته می‌شود. با کمی دقت به قسمت شرط این فلوجارت متوجه می‌شوید که در صورت برقراری شرط، دو دستور باید اجرا شود. در حالی که در مثال‌های قبلی تنها یک دستور اجرا می‌شد. برای مشخص کردن این دو دستور در برنامه، باید آنها را در بین علامت‌های آکولاد باز و بسته قرار دهیم. با این کار یک بلاک می‌سازیم.

```
{
    دستور محاسبه مالیات
    دستور نمایش مقدار مالیات
}
```

بلاکی شامل دو دستور ←

بلاک چیست؟

به تعدادی دستور که داخل علامت‌های آکولاد باز و بسته قرار داشته باشند بلاک گفته می‌شود.

```
{
    دستور ;
    دستور ;
    دستور ;
}
```

● بلاک می‌تواند خالی باشد یعنی بین علامت‌های آکولاد، هیچ دستوری وجود نداشته باشد.

● بلاک می‌تواند فقط شامل یک دستور باشد.

● برای خوانا و واضح شدن یک بلاک، دستورهای داخل بلاک را با تورفتگی

می‌نویسیم تا به راحتی در برنامه مشخص شوند.



فلوچارت ۷-۶- محاسبه حقوق دریافتی کارمندان یک شرکت

```

class Salary
{
    static void Main()
    {
        long income, tax, insurance, net;
        string input;
        Console.WriteLine("Enter income: ");
        input = Console.ReadLine();
        income = long.Parse(input);
        insurance = income * 0.07;
        if (income > 10000000)
        {
            tax = (income - 10000000) * 0.05;
            Console.WriteLine("Tax " + tax);
        }
        else
            tax = 0;
        Console.WriteLine("Insurance " + insurance);
        net = income - insurance - tax;
        Console.WriteLine("Net " + net);

        Console.WriteLine("Press any key to continue...");
        Console.ReadKey();
    }
}

```

برنامه ۶-۴- محاسبه حقوق خالص کارمندان یک شرکت (مثال ۶-۷)

سؤال: در مثال ۶-۷، برای محاسبه مالیات، ابتدا تفاوت حقوق با ده میلیون ریال محاسبه شده و سپس در پنج درصد ضرب شده است. به این روش محاسبه، مالیات مازاد گفته می‌شود. به نظر شما چرا مستقیماً حقوق در پنج درصد ضرب نمی‌شود؟

۵-۶ عملگرهای منطقی

عملگر منطقی && یک عملگر دوتایی و دارای دو عملوند است. در هنگام اجرای آن، ابتدا نتیجه عملوند سمت چپ، به وسیله کامپیوتر محاسبه و درستی یا نادرستی آن مشخص می‌شود. اگر ارزش عملوند سمت چپ false باشد نتیجه عملگر && نیز false خواهد بود. اما اگر نتیجه عملوند سمت چپ true باشد، آنگاه عملوند سمت راست محاسبه می‌شود که بر اساس ارزش آن، نتیجه عملگر && طبق جدول ۶-۲ مشخص می‌شود.

جدول ۶-۲ - نتیجه عملگر &&

ارزش عملوند سمت چپ	ارزش عملوند سمت راست	نتیجه عملگر &&
false	false	false
false	true	false
true	false	false
true	true	true

مثال ۸-۶: فرض کنید نمره درسی یک دانش آموز در متغیر mark قرار دارد. با بررسی قبولی یا مردودی، پیام مناسب چاپ کنید.

الگوریتم یا روش انجام کار: در این مثال دو محدوده از اعداد، باید بررسی شود. بنابراین به دو دستور if نیاز داریم که در صورت درست بودن عبارت منطقی مربوط به هر یک از آنها، پیامی بر روی صفحه نمایش داده شود. برای این مثال از فلوجارت استفاده نمی‌کنیم و به جای آن مستقیماً دستورها را می‌نویسیم. شکل زیر ساختار دستورها را نشان می‌دهد. عبارت‌های داخل پرانتز به زبان فارسی است که باید به زبان C# نوشته شوند:

(نمره کمتر از ده) if

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

(نمره از ده تا بیست) if

```
Console.WriteLine("Passed"); // قبول
```

عبارت (نمره کمتر از ده) در زبان C#، به صورت عبارت منطقی ($mark < 10$) نوشته می‌شود در این عبارت ساده در صورتی که نمره (mark) عددی کمتر از ده، مثلاً ۸ باشد نتیجه عبارت true است و در نتیجه دستور WriteLine() اجرا شده و پیام «Failed» را نمایش می‌دهد.

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

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

اما عبارت (نمره از ده تا بیست) به سادگی عبارت قبلی در زبان C# نوشته نمی‌شود، بلکه این عبارت را باید با ترکیب دو عبارت «نمره بزرگتر یا مساوی ۱۰» و همچنین «نمره کوچکتر یا مساوی ۲۰» بسازید.

```
(mark > 10) و (mark < 20)
```

بین دو عبارت، «و» قرار دارد که برای ترکیب دو عبارت منطقی استفاده می‌شود، به طوریکه هر گاه نتیجه هر دو عبارت true باشد، نتیجه کل عبارت نیز درست است ولی اگر نتیجه حداقل یکی از عبارت‌ها false باشد، نتیجه کل عبارت نیز نادرست است. «و» در واقع یک عملگر منطقی است. بنابراین ترکیب دو عبارت قبلی با استفاده از عملگر && به صورت زیر است:

```
(mark > 10) && (mark < 20)
```

در این عبارت مرکب، به ازای 14 mark، نتیجه هر دو عبارت سمت چپ و راست، برابر true است. بنابراین نتیجه کل عبارت نیز true است. اما به ازای 9 mark نتیجه عبارت false ($mark > 10$) است بنابراین نتیجه کل عبارت نیز false خواهد بود.

حال دستور if مربوط به وضعیت قبولی دانش آموز چنین خواهد بود:

```
if ((mark > 10) && (mark < 20))
```

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

دستورهای کامل به صورت زیر است:

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

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

```
if ((mark > 10) && (mark < 20))
```

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

علاوه بر عملگر &&، عملگرهای دیگری نیز در زبان C#، برای ایجاد عبارت‌های مرکب وجود دارد، که علامت و عملکرد آنها به ترتیب اولویت، در جدول ۳-۶ نشان داده شده است.

جدول ۳-۶- عملگرهای منطقی

اولویت	نام عملگر	علامت	عملکرد
۱	نقیض	!	ارزش عملوند را معکوس می کند
۲	و	&&	تنها در صورتی که هر دو عملوندها true باشند نتیجه این عملگر نیز true خواهد بود در غیر این صورت false است
۳	یا		اگر حداقل یکی از عملوندها true باشند نتیجه این عملگر نیز true خواهد بود
۴	یا انحصاری ^۱	^	اگر ارزش عملوندها مخالف یکدیگر باشند، نتیجه این عملگر true خواهد بود

جدول ۴-۶- حالت های مختلف در عملگرهای منطقی

نتیجه عملگر ^	نتیجه عملگر	ارزش عبارت سمت راست	ارزش عبارت سمت چپ
false	false	false	false
true	true	true	false
true	true	false	true
false	true	true	true

کار در کارگاه ۳

مثال ۹-۶: می خواهیم برنامه ای بنویسیم که عددی را دریافت کند و تشخیص دهد که مضرب

پنج است یا خیر.

الگوریتم یا روش کار: برای تشخیص اعدادی که مضرب پنج هستند، از دو روش می توان

استفاده می شود. یک روش، بررسی بخش پذیری بر پنج است، مانند روشی که برای تشخیص اعداد

زوج در مثال ۵-۶، استفاده کردیم. روش دیگر، بررسی رقم یکان عدد است که در این مثال، می خواهیم

از آن استفاده کنیم. در این روش، رقم یکان عدد را جدا می کنیم. اگر رقم یکان برابر پنج یا برابر صفر

بود، آن گاه عدد، مضرب پنج است. به عبارت مرکب در دستور if، توجه کنید که از عملگر استفاده

شده است.

^۱ _Exc us ve or (XOR)

```

class CheckNumbers
{
    static void Main()
    {
        string input;
        int number;

        Console.WriteLine("Enter a number: ");
        input = Console.ReadLine();
        number = int.Parse(input);

        int firstDigit = number % 10;
        if ((firstDigit == 0) || (firstDigit == 5))
            Console.WriteLine("The number is a multiple of 5.");

        Console.WriteLine("Press any key to continue...");
        Console.ReadKey();
    }
}

```

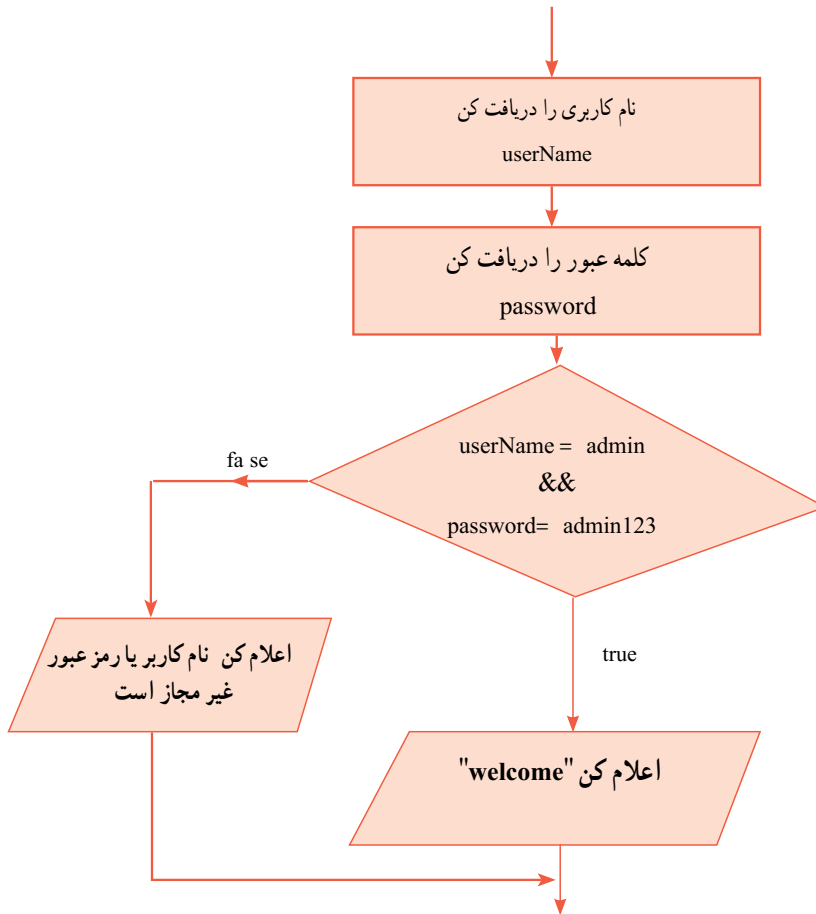
برنامه ۵-۶- برنامه تشخیص عدد مضرب پنج (مثال ۹-۶)

سؤال؟ خروجی برنامه به ازای اعداد ورودی ۹ و ۳۷ چیست؟ چه پیامی روی صفحه ظاهر می‌شود؟

سؤال؟ چه دستورهایی به برنامه اضافه می‌کنید، تا به ازای اعداد ورودی غیر مضرب ۵ مانند ۳۷، خروجی برنامه واضح باشد؟

مثال ۱-۶: می‌خواهیم مثال ۳-۶ را تکمیل کنیم به صورتی که شخص خاصی بتواند از آن استفاده نماید. بنابراین با اجرای برنامه، باید نام کاربری و کلمه رمز دریافت شده و درستی ورودی‌ها در برنامه بررسی شود.

الگوریتم یا روش کار: در این مثال، ابتدا نام کاربری و کلمه عبور را از کاربر دریافت کرده و آنها را در دو متغیر ذخیره می‌کنیم. سپس محتوای متغیرها را با نام کاربری و کلمه عبور از قبل تعریف شده، مقایسه می‌نماییم. اگر اطلاعات وارد شده، صحیح بود پیام خوشامد گویی نمایش داده می‌شود. در غیر این صورت، برنامه پیام می‌دهد که اطلاعات وارد شده صحیح نیست (فلوچارت ۶-۸).



فلوچارت ۶-۸- دریافت نام کاربری و کلمه عبور

در فلوچارت ۶-۸، فرض شده است که کاربر مجاز، دارای نام کاربری «admin» و کلمه عبور «admin123» است. برای بررسی اطلاعات وارد شده از یک عبارت منطقی مرکب، در دستور if-else استفاده می‌کنیم. برنامه شماره ۶-۶ بر اساس این فلوچارت نوشته شده است:

```

class Login
{
static void Main()
{
string userName, password;

Console Write("Enter username: ");
userName = Console ReadLine();

Console Write("Enter password: ");
password = Console ReadLine();

if((userName == "admin") && (password == "admin123"))
Console WriteLine("Welcome Admin ");
else
Console WriteLine("Invalid username or password! ");

Console WriteLine("Press any key to continue ");
Console ReadKey();

}
}

```

برنامه ۶-۶- دریافت نام کاربری و کلمه عبور کاربر (مثال ۱۰-۶)

مثال ۱۱-۶: می‌خواهیم برنامه‌ای بنویسیم که با توجه به عدد دریافتی از کاربر، نام یکی از فصل‌های سال را نمایش دهد (۱: بهار، ۲: تابستان، ...). الگوریتم و روش انجام کار: ابتدا عدد دریافتی را در متغیر number قرار می‌دهیم. سپس با مقایسه عدد دریافتی با شماره‌های ۱ تا ۴ نام فصل مربوطه را نمایش می‌دهیم.

۶-۶ ساختار if_else پیچیده

در این ساختار در صورتی که عبارت منطقی ۱ درست (true) نباشد، عبارت منطقی ۲ بررسی می‌شود و در صورتی که هیچ یک از عبارات منطقی ۱، ۲ و ۳ درست (true) نباشد، دستورات شماره ۴ انجام می‌شود.

```
if (عبارت منطقی ۱)
{
    دستورات شماره ۱ ;
}
else if (عبارت منطقی ۲)
{
    دستورات شماره ۲ ;
}
else if (عبارت منطقی ۳)
{
    دستورات شماره ۳ ;
}
else
{
    دستورات شماره ۴ ;
}
```

در مثال ۶-۱۱ چنین ساختاری مورد استفاده قرار می‌گیرد.

```

class CheckSeason
{
    static void Main(string[] args)
    {
        string input;
        int number;
        Console Write("Enter a number: ");
        input = Console ReadLine();
        number = int Parse(input);

        if (number == ۱)
            Console WriteLine("Spring");
        else if (number == ۲)
            Console WriteLine("Summer");
        else if (number == ۳)
            Console WriteLine("Fall");
        else if (number == ۴)
            Console WriteLine("Winter");
        else
            Console WriteLine("Invalid Number! ");

        Console WriteLine("Press any key to continue ");
        Console ReadKey();
    }
}

```

برنامه ۶-۶- دریافت شماره فصل و نمایش نام فصل (مثال ۱۱-۶)

۷-۶- دستور switch

در مواردی که بخواهیم حالت‌های مختلف یک عبارت را بررسی و بر اساس آن دستورهای را اجرا کنیم، از دستور switch استفاده می‌کنیم. ساختار کلی این دستور را می‌بینید.

```
switch (عبارت)
{
    مقدار ۱: case
        دستور ۱;
        break;
    مقدار ۲: case
        دستور ۲;
        break;
    .
    .
    default:
        دستورهای دیگر;
        break;
}
```

ساختار کلی دستور switch

در جلوی کلمه رزرو شده switch، عبارتی در داخل پرانتز نوشته می‌شود که بر اساس حاصل آن، تصمیم‌گیری و اجرای دستورها کنترل می‌شود. مقادیری که حاصل عبارت با آن‌ها مقایسه می‌شود، هر یک در جلوی کلمه رزرو شده case نوشته می‌شود. اگر حاصل عبارت با یک مقدار case برابر باشد، آن‌گاه دستور یا دستورهای جلوی case تا رسیدن به کلمه رزرو شده break اجرا می‌شود. اگر حاصل عبارت با هیچ کدام از مقادیر case برابر نشود، آن‌گاه دستورهای قسمت default اجرا می‌شوند.

آکولادهای باز و بسته محدوده عملیات شروع و پایان دستور switch را معین می‌کند. نوع عبارتی که داخل پرانتز دستور switch نوشته می‌شود، نمی‌تواند اعراساری باشد. اما عبارت‌های حرفی، رشته‌ای و انواع داده صحیح می‌تواند استفاده شود.

مثال ۶-۱۲: اکنون می‌خواهیم مثال ۶-۱۱ را با استفاده از دستور switch بازنویسی نماییم.

```
class CheckSeason
{
    static void Main()
    {
        string input;
        Console.WriteLine("Enter a number: ");
        input = Console.ReadLine();
        switch (input)
        {
            case "1":
                Console.WriteLine("Spring");
                break;
            case "2":
                Console.WriteLine("Summer");
                break;
            case "3":
                Console.WriteLine("Fall");
                break;
            case "4":
                Console.WriteLine("Winter");
                break;
            default:
                Console.WriteLine("Invalid Number!");
                break;
        }
        Console.WriteLine("Press any key to continue...");
        Console.ReadKey();
    }
}
```

برنامه ۶-۷— دریافت شماره فصل و نمایش نام فصل (مثال ۶-۱۲)

۱- مثال ۶-۱۳: در این جا می‌خواهیم مثال ۲-۶ را توسعه دهیم. تشخیص سه کاربر مختلف، با دریافت نام کاربری و کلمه عبور، انجام شود. در این برنامه، نام کاربری با دستور switch و کلمه عبور با دستور if بررسی شده است.

```
class SwitchDemo
{
    static void Main(string[] args)
    {
        string userName, password;
        Console.Write("Enter username: ");
        userName = Console.ReadLine();
        Console.Write("Enter password: ");
        password = Console.ReadLine();
        switch (userName)
        {
            case "admin":
                if (password == "admin123")
                    Console.WriteLine("Welcome Manager.");
                else
                    Console.WriteLine("Wrong password!");
                break;
            case "accountant":
                if (password == "acc123")
                    Console.WriteLine("Welcome accountant.");
                else
                    Console.WriteLine("Wrong password!");
```

```

        break;
    case "sales":
        if (password == "sales123")
            Console.WriteLine("Welcome Manager!");
        else
            Console.WriteLine("Wrong password!");
        break;
    default:
        Console.WriteLine("Invalid username!");
        break;
    }
    Console.WriteLine("Press any key to continue . . .");
    Console.ReadKey();
}
}

```

برنامه ۶-۷ — دریافت نام کاربری و رمز عبور کاربران (مثال ۱۳-۶)

اگر به ساختار دستور switch توجه کنید، پس از هر دستور case، یک دستور break نوشته شده است. از کلمه رزرو شده break برای خاتمه دادن به یک case استفاده می‌شود. اگر دستور break نوشته نشود، مترجم برای جلوگیری از اشتباه برنامه‌نویس، خطا می‌دهد^۱. برای هر case، می‌توان بیش از یک دستور نوشت و نیازی به بلاک ندارد.

۲- مثال ۱۴-۶: با استفاده از قطعه برنامه زیر، متن کامل برنامه را در محیط VS بنویسید. در این قطعه برنامه از کاربر سؤالی پرسیده می‌شود، کاربر در پاسخ به سؤال، کلمه‌ای را وارد می‌کند. اگر کاربر کلمه Yes و یا کلمه maybe را وارد نماید، هر دو یک نتیجه را خواهد داشت و پیام Great! بر روی صفحه نشان داده می‌شود.

۱- در زبان C یا C، می‌توانید دستور break را بنویسید که در این صورت بعد از پایان اجرای دستورات یک case، دستورات

case بعدی نیز اجرا می‌شود.

```

Console.WriteLine("Do you enjoy C# ? (yes/no/maybe): ");
string input = Console.ReadLine();
switch (input)
{
    case "yes":
    case "maybe":
        Console.WriteLine("Great!");
        break;
    case "no":
        Console.WriteLine("Too bad!");
        break;
}

```

قطعه برنامه ۸-۶- کاربرد دستور switch بدون break

۳- برنامه‌ای بنویسید که یک عدد را بگیرد و زوج یا فرد بودن آن را تشخیص دهد.

۴- برنامه‌ای بنویسید که نمرهٔ دانش‌آموز را دریافت کند و پیامی برای قبولی یا مردودی ا صادر نماید.

۵- برنامه شماره ۱۱-۶ و ۱۲-۶ را نوشته و ترجمه کنید. به ازای ورودی‌های مختلف آن را

اجرا کنید.

۶- اگر کاربر در هنگام اجرای مثال ۱۴-۶، کلمه دیگری به غیر از کلمات تعیین شده بنویسد،

خروجی چه خواهد بود؟

۷- می‌خواهیم اگر کاربر کلمه دیگری غیر از کلمات تعیین شده در مثال ۱۳-۶، را وارد کرد،

دستور زیر اجرا گردد، چه تغییری در داخل دستور switch انجام می‌دهید؟

```

Console.WriteLine("I'm sorry, I don't understand that!");

```

خودآزمایی فصل ششم

۱- دستورهای زیر برای کنترل محتوای number نوشته شده است. با تغییراتی در عبارت منطقی و دستور if، همین کنترل را با استفاده از عملگر && انجام دهید:

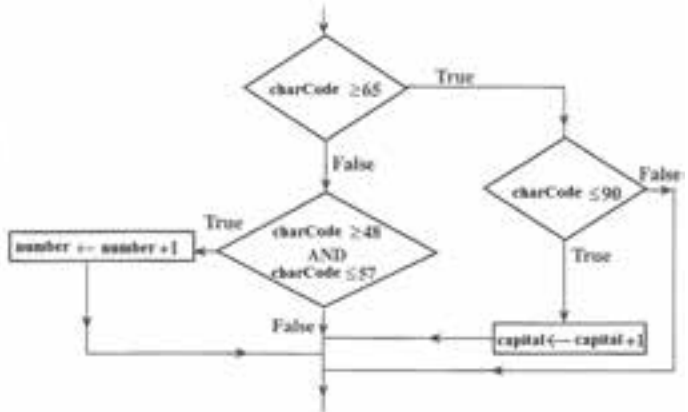
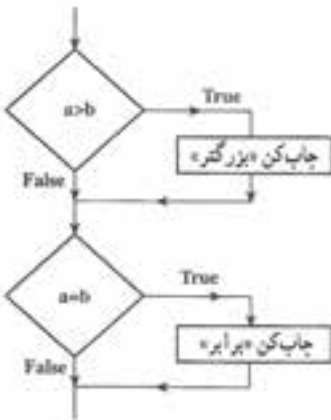
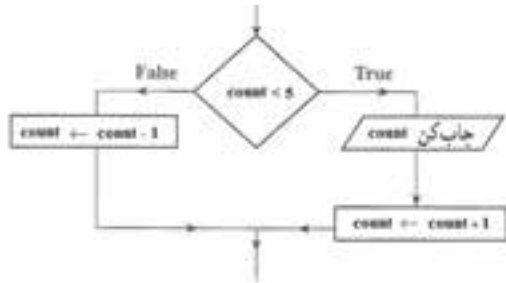
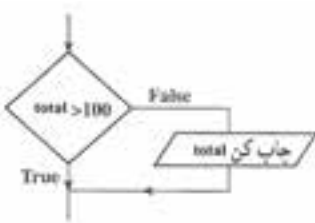
```
if((number > 10) (number < 0))
```

```
    Console.WriteLine("The number should between zero to ten.");
```

```
else
```

```
    Console.WriteLine("Good job!");
```

۲- دستورهای شرطی مطابق با فلوجارت‌های زیر بنویسید:



۳- بدون استفاده از دستور if، دستور زیر را بنویسید.

```
if (number < 100)
```

```
    number ;
```

```
else
```

```
    number 1;
```


۴- در هر یک از دستوره‌های زیر، با استفاده از عملگرهای منطقی، تعداد دستوره‌های if را کاهش دهید.

الف)

```
if (score > 0)
    if (score < 100)
        Console.WriteLine("Score " + score);
```

ب)

```
if (choice == 1)
    Console.Clear();
if (choice == 2)
    Console.Clear();
Console.WriteLine("Stating...");
```

۵- اگر در متغیر char، یک کاراکتر قرار داشته باشد، با استفاده از دستوره‌های شرطی معین کنید که در داخل متغیر، چه نوع کاراکتری (حروف بزرگ، حروف کوچک و یا ارقام) قرار دارد؟

۶- در هر یک از دستوره‌های شرطی زیر، دستور WriteLine() در چه صورت اجرا می‌شود؟

الف)

```
if (number > 0)
    if (number < 10)
        failed ;
    else
        Console.WriteLine(number);
```

ب)

```
if (number > 0)
{
    if (number < 10)
        failed ;
}
else
    Console.WriteLine(number);
```

تمرینات برنامه‌نویسی فصل ششم

- ۱- برنامه‌ای بنویسید که عددی را به عنوان شماره فصل از ورودی دریافت کند. نام ماه‌های آن فصل را در خروجی چاپ نماید. برای ورودی غیر مجاز، پیام مناسب نشان داده شود.
- ۲- برنامه‌ای بنویسید که سن کاربر را دریافت کند و تعداد روزهای عمر او را نشان دهد. عدد وارد شده توسط کاربر در برنامه کنترل شود. سن کاربر نمی‌تواند منفی و یا بزرگتر از عددی مانند ۱۵۰ باشد.
- ۳- برنامه‌ای بنویسید که عددی را از ورودی دریافت کند و تعیین کند که این عدد مضرب ۶ است یا خیر؟ از روش بخش‌پذیری هم زمان بر ۲ و ۳ استفاده نمایید.
- ۴- برنامه‌ای بنویسید که عددی را از ورودی دریافت کند و زوج بودن آن را تشخیص دهد. برای بخش‌پذیری عدد ورودی بر ۲، از روش بررسی رقم یکان شامل ۰، ۲، ۴، ۶، ۸ استفاده کنید.
- ۵- برنامه‌ای بنویسید که یک عدد یک رقمی دریافت کند و کلمه متناظر با آن عدد را روی صفحه نمایش دهد مثلاً اگر عدد ۵ از ورودی دریافت شد، برنامه کلمه FIVE را نمایش دهد. این برنامه را برای عدد دو رقمی توسعه دهید.
- ۶- برنامه‌ای بنویسید که میزان مصرف برق در یک ماه را بر حسب کیلو وات سؤال نماید و سپس بهای برق مصرفی یک ماه را، بر اساس جدول زیر محاسبه کند و مبلغ آن را بر حسب ریال نمایش دهد. مثلاً اگر مصرف برق ۱۸۰ کیلو وات در ماه باشد در این صورت ۱۰۰ کیلو وات آن با مبلغ ۱۳۶۴ ریال محاسبه شده و ۸۰ کیلو وات باقیمانده آن، با مبلغ ۱۴۲۶ ریال محاسبه می‌شود:

$$(۱۰۰ \times ۱۳۶۴) + (۸۰ \times ۱۴۲۶)$$

پله‌های مصرف ۳۰ روزه (kwh)	نرخ (ریال)
مصرف تا ۱	۱۳۶۴
مازاد بر ۱ تا ۲	۱۴۲۶
مازاد بر ۲ تا ۳	۱۴۸۸
مازاد بر ۳ تا ۴	۱۵۵
مازاد بر ۴ تا ۵	۱۷۳۶
مازاد بر ۵ تا ۶	۱۹۸۴
مازاد بر ۶	۲۲۳۲

- ۷- برنامه‌ای بنویسید که سه عدد از ورودی دریافت کند و عدد بزرگتر را تشخیص داده و آن را نمایش دهد.

۸- برنامه‌ای بنویسید که شماره روز را دریافت و نام روز را چاپ کند. مثلاً اگر اولین روز سال جاری چهارشنبه بوده است، با ورود عدد ۱، چهارشنبه، با ورود عدد ۲، پنج‌شنبه و ... چاپ شود. از دستور switch استفاده کنید.

۹- برنامه‌ای بنویسید که دمای هوا را از ورودی دریافت نماید. بر طبق جدول زیر، پیام مناسب را چاپ نماید.

ورودی	خروجی
$\text{دما} < ۱$	بسیار سرد
$۱ \leq \text{دما} < ۲$	سرد
$۲ \leq \text{دما} < ۳$	معتدل
$۳ \leq \text{دما} < ۴$	گرم
$\text{دما} > ۴$	بسیار گرم

۱۰- برنامه‌ای بنویسید که شماره رنگ را از ورودی دریافت نماید. طبق جدول زیر، رنگ زمینه صفحه نمایش را به رنگ مربوطه تغییر دهد.

ورودی (شماره رنگ)	رنگ صفحه نمایش
	مشکی
۱	آبی
۲	قرمز
۳	سبز

۱۱- فرض کنید آزمون‌های دارای ۳۰ سؤال چهارگزینه‌ای باشد. برنامه‌ای بنویسید که تعداد پاسخ‌های درست و تعداد پاسخ‌های نادرست یک شرکت‌کننده در این آزمون را دریافت کرده، نمره وی و درصد پاسخ‌گویی به سؤالات را محاسبه نماید. (با این فرض که هر پاسخ غلط $\frac{1}{3}$ نمره منفی دارد). مثلاً اگر دانش‌آموز ۱۷ پاسخ درست و ۶ پاسخ غلط داشته باشد، نمره وی برابر ۱۵ می‌باشد (از ۳۰ نمره) و درصد پاسخ‌گویی وی، برابر ۵۰٪ است.

۱۲- آیا برنامه‌ی تمرین ۱۱، کنترلی بر روی صحت داده‌های ورودی برنامه دارد؟ مثلاً کنترل می‌کند که مجموع تعداد پاسخ‌های صحیح و غلط کمتر یا مساوی تعداد کل سؤالات آزمون باشد؟

۱۳- با تکمیل تمرین ۱۱ از بروز چنین خطاهایی در ورودی جلوگیری کنید.