



پودمان ۵

حل مسائل شرطی



استفاده از رایانه برای تولید برنامه و انجام محاسبات ریاضی و مسائل مختلف کاری امری عادی شده است. یک مسئله در دنیای واقعی دارای حالت‌های گوناگونی است که در صورت وقوع هر یک باید تصمیم و پاسخ مناسبی برای آن ایجاد شود. وقتی بخواهیم برای این مسائل مربوط به دنیای واقعی برنامه تولید کنیم، ضمن استفاده از زبان‌های برنامه نویسی باید شرایط و حالت‌های مختلف را نیز در برنامه در نظر بگیریم. برای بررسی این حالت‌ها در زبان‌های برنامه‌نویسی از ساختارهای شرطی و کنترل‌های خطا استفاده می‌شود. در این پودمان هنرجو قادر خواهد بود با اتکا بردانش انواع عملگرها و ساختارهای شرطی و مهارت به‌کارگیری این عملگرها و تشخیص وجود ساختار شرط در برنامه به‌وسیله تحلیل الگوریتم برنامه، برنامه‌های شرطی تولید کرده، خطاهای آن را برطرف کند.

واحد یادگیری ۹

شایستگی کار با عملگرهای ریاضی، مقایسه‌ای و منطقی

آیا تا به حال پی برده اید

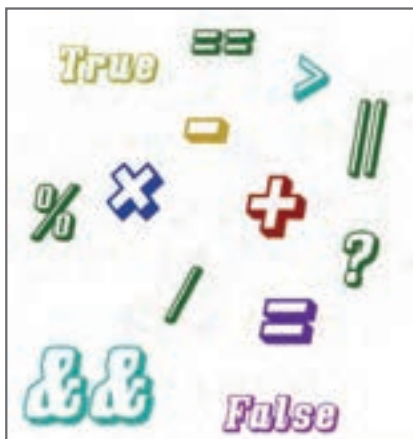
- عملگرهای ریاضی در زبان برنامه‌نویسی سی شارپ چگونه نمایش داده می‌شوند؟
- چگونه عبارات ریاضی و مقایسه‌ای محاسبه می‌شوند؟
- حاصل یک عبارت مقایسه‌ای از چه نوعی است؟
- در یک عبارت محاسباتی اولویت انواع عملگرها نسبت به یکدیگر چگونه است؟
- کدام عملگر ریاضی می‌تواند روی داده‌های غیر عددی نیز عمل کند؟

هدف از این واحد شایستگی به کارگیری انواع عملگرها برای ایجاد عبارات محاسباتی در برنامه و محاسبه مقدار یک عبارت محاسباتی با توجه به اولویت عملگرها است.

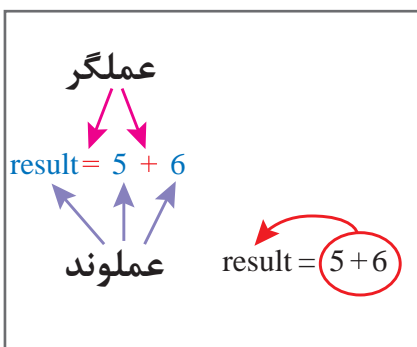
استاندارد عملکرد

با استفاده از دانش ریاضی و شناخت انواع عملگرها و اولویت آنها نسبت به یکدیگر در زبان برنامه‌نویسی، عبارات محاسباتی ایجاد و نتیجه آنها را محاسبه کرده، خطاهای احتمالی آن را برطرف کند.

پودمان پنجم: حل مسائل شرطی



یکی از کاربردهای اولیه رایانه، انجام عملیات ریاضی و مقایسه‌ای است. در زبان‌های برنامه‌نویسی برای انجام این عملیات از عبارت (expression) استفاده می‌شود. به عبارت $result=5+6$ دقت کنید.



یک عبارت حداقل از یک عملگر (operator) و یک یا چند عملوند (operand) تشکیل شده است. به اعدادی مانند ۵ و ۶ و متغیر $result$ که یک عملگر روی آنها عملی را انجام می‌دهد، عملوند می‌گویند. به علامت‌هایی مانند علامت + و = که بیانگر انجام یک عمل روی متغیرها و داده‌ها هستند، عملگر می‌گویند. هر عبارت دارای حاصل و نتیجه‌ای است که می‌تواند به یک متغیر انتساب داده شود.

متغیر $result$ از چه نوعی می‌تواند باشد؟

کنجکاوی



زبان C# از زبان‌های سطح بالاست و از عملگر + برای جمع و عملگر = برای انتساب استفاده می‌کند.

در جدول ۱ عملوندها و عملگرهای عبارات محاسباتی را مشخص کنید.

فعالیت کارگاهی



جدول ۱- شناسایی عملگر و عملوند در عبارت

عبارت	عملگر	عملوند
-5		
7-13+12		
a-b*2		

می‌توان حاصل عبارت را با متد Write یا WriteLine به صورت مستقیم در خروجی نمایش داد.

`Console.WriteLine(6+5);`

دستور `Console.WriteLine(6+5);` چه مقداری را روی صفحه نمایش نشان می‌دهد؟

کنجکاو



یادداشت



عملوندهای عبارت محاسباتی می‌توانند داده یا متغیر باشند. `result = num1 + num2;`

عملگرهای محاسباتی

در زبان‌های برنامه‌نویسی برای انجام عملیات ریاضی پایه شامل جمع، تفریق، ضرب و تقسیم عملگرهایی وجود دارد. جدول ۲ این عملگرها را در زبان سی‌شارپ نشان می‌دهد.

جدول ۲ را تکمیل کنید.

فعالیت
کارگاهی



جدول ۲ - عملگرهای محاسباتی در سی‌شارپ

عملگر	عملکرد	مثال	نتیجه
-	قرینه	-5	
*	ضرب	20*6	
/	تقسیم	4/25	
%	باقیمانده تقسیم	6%20	
+	جمع	3+20	
-	تفریق	3-20	

به عملگرهایی مانند عملگر قرینه که به یک عملوند نیاز دارند، عملگرهای **یکتایی** (Unary) و به عملگرهایی مانند عملگرهای + و * و / که به دو عملوند نیاز دارند، عملگرهای **دوتایی** (Binary) می‌گویند.

کارگاه ۱ | عملگر تقسیم

بُعد خانوار از جمله شاخص‌های با اهمیت یک کشور است که نشان‌دهنده میانگین تعداد افراد یک خانواده در آن کشور است. می‌خواهیم با توجه به آمار سال ۹۰ کشورمان، برنامه‌ای بنویسیم که جمعیت مردم ایران و تعداد خانوار را دریافت کرده، بعد خانوار را در ایران نمایش دهد.

1 پروژه جدید به نام `ConsloePopulation` ایجاد کنید.

پودمان پنجم: حل مسائل شرطی

۲ کد زیر را در متد Main بنویسید.

```
Console.WriteLine("----- family size in IRAN -----");
```

```
Console.WriteLine("Enter Population:");  
long populate = long.Parse(Console.ReadLine());
```

```
Console.WriteLine("Enter number of family :");  
long numFamily = long.Parse(Console.ReadLine());  
float familySize = populate / numFamily;  
Console.WriteLine("family size in IRAN: {0}", familySize);
```

۳ برنامه را با مقادیر داده شده در جدول اجرا کرده، آن را تکمیل کنید.

سال	جمعیت	تعداد خانوار	خروجی	جواب شما
۹۰ (کشور)	۷۵۱۴۹۶۶۹	۲۱۱۸۵۶۴۷		
۹۰ (کاشان)	۳۲۳۳۷۱	۹۵۳۱۴		
۹۰ (فارس)	۹۳۹۴۱	۲۴۷۶۵		

آیا خروجی برنامه با جواب شما برای بعد خانوار، یکسان است؟

نتیجه عملگر تقسیم با توجه به نوع عملوندهایش می‌تواند صحیح و یا اعشاری باشد. اگر حداقل یکی از عملوندهای عملگر تقسیم، از نوع داده اعشاری باشد، نتیجه عبارت، عدد اعشاری و در غیر این صورت عدد صحیح است.

۴ برای حل مشکل برنامه، عبارت محاسبه familySize را به صورت زیر تغییر دهید:

```
float familySize = (float) populate/ numFamily;
```

۵ برنامه را دوباره اجرا کرده، جدول زیر را تکمیل کنید.

سال	جمعیت	تعداد خانوار	خروجی	جواب شما
۹۰ (کشور)	۷۵۱۴۹۶۶۹	۲۱۱۸۵۶۴۷		
۹۰ (زادگاه شما)				
۹۰ (استان همسایه)				

۶ آمار جمعیت زادگاه خود و استان همسایه را از آمار سال ۹۰ در جدول بالا قرار داده، بعد خانوار را در دو شهر باهم مقایسه کنید.



برای حل مشکل برنامه، چه روش‌های دیگری پیشنهاد می‌کنید؟



- برنامه‌ای بنویسید که سن شما را از ورودی خوانده، تعیین کند چندماه، چندروز و چندساعت از عمر شما می‌گذرد؟
 - در برنامه بالا سن دریافتی را ۷ برابر کرده، حاصل را ۱۴۴۳ برابر کنید و نتیجه را نمایش دهید. نتیجه را با سن دریافتی مقایسه کنید.
 - برنامه را برای سن برادر، خواهر و یا هم‌کلاسی خود دوباره اجرا کنید.
 - برنامه‌ای بنویسید که مدت زمان لازم برای انجام این فعالیت را برحسب ثانیه دریافت کند و تعیین کند چند دقیقه و ثانیه برای انجام این فعالیت فرصت دارید؟



آنچه آموختم:

- ۱.....
- ۲.....
- ۳.....

اولویت‌بندی عملگرهای محاسباتی

$$3+2*10$$

نتیجه محاسبه عبارت روبه‌رو چیست؟

این عبارت دارای دو عملگر + و * است. کدام عملیات اول انجام می‌شود؟ ۲ حالت وجود دارد.

الف) ابتدا جمع و بعد عملگر ضرب انجام شود که نتیجه ۵۰ خواهد بود.

$$1) 3+2=5$$

$$2) 5*10=50$$

$$3 + 2 * 10$$

۱
۲

↓
↓

ب) ابتدا ضرب و سپس جمع انجام شود که نتیجه ۲۳ خواهد بود.

$$1) 2*10=20$$

$$2) 3+20=23$$

$$3 + 2 * 10$$

۲
۱

↓
↓

- برنامه‌ای بنویسید که حاصل عبارت $3+2*10$ را نمایش دهد. کدام یک از نتایج بالا نمایش داده می‌شود؟
 - در برنامه بالا عبارت $10*(3+2)$ را جایگزین کنید و نتیجه را بنویسید.



پودمان پنجم: حل مسائل شرطی

جدول ۳ اولویت عملگرهای محاسباتی را نشان می‌دهد. چنانچه چند عملگر اولویت یکسانی داشته باشند، عملگرها به ترتیب از چپ به راست انجام می‌شوند و به آن "شرکت پذیری چپ" می‌گویند.

جدول ۳- اولویت عملگرهای محاسباتی

اولویت	عملگر	نوع عملگر
۱	-	یکتایی
۲	*	دوتایی
	/	
	%	
۳	+	دوتایی
	-	

با توجه به جدول اولویت‌ها در سی‌شارپ (جدول ۳)، جدول زیر را تکمیل کرده، اولویت عملگرها را تعیین کنید.

نتیجه عبارت	مقدار متغیرها	معادل در #C	عبارت ریاضی	شرح عبارت
	b=2 a=1 c=1	$\Delta = b * b - 4 * a * c$	$\Delta = b^2 - 4ac$	محاسبه دلتا برای حل معادله درجه ۲
		$(a+b+c)/3$		میانگین سه عدد
				محاسبه حقوق خالص با کسر بیمه ۳ درصد
				تعداد گروه‌های سه نفره در کارگاه با توجه به تعداد هنرجویان
				محاسبه معدل هنرجو با توجه به تعداد واحدهای هر درس

فعالیت کارگاهی



در تبدیل یک عبارت ریاضی به معادل آن در سی‌شارپ، درج پرانتز در چه مواقعی ضروری است؟

کنجکاوی





در صورتی که در یک عبارت محاسباتی چندین پرانتز تودرتو وجود داشته باشد؛ محاسبات از داخلی‌ترین پرانتز آغاز می‌شود.

کارگاه ۲ تغییر اولویت عملگرها

می‌خواهیم برای فروشگاه‌ای که دارای اجناس تخفیف‌دار است، برنامه‌ای بنویسیم. ۳ کالای اولی که خریداری شده، دارای تخفیف ۱۰ درصد و تخفیف کالای چهارم ۷۰ درصد است. برنامه‌ای بنویسید که قیمت قبل از تخفیف ۴ کالا را از ورودی دریافت کرده، مبلغ تخفیف را محاسبه کند.

۱ پروژه جدیدی به نام `ConsoleDiscount` ایجاد کنید.

۲ قطعه کد زیر برای دریافت قیمت کالاها از ورودی است. آن را تکمیل کنید.

```
Console.WriteLine (" ----- Final Price-----");
```

```
//Get the First Price
```

```
Console.Write ("\nEnter Price1:");
```

```
float price1 = .....
```

```
//Get the Second Price
```

```
Console.Write ("\nEnter Price2:");
```

```
float price2 = .....
```

```
//Get the Third Price
```

```
Console.Write ("\nEnter Price3:");
```

```
float price3 = .....
```

```
//Get the fourth Price
```

```
Console.Write ("\nEnter Price4:");
```

```
float price4 = .....
```

۳ دستورات زیر مبلغ نهایی را محاسبه کرده، نمایش می‌دهد. این دستورات را به کدهای قبلی اضافه کنید.

```
//Calculate the Discount
```

```
float discount = price1 + price2 + price3* 10 / 100 + price4* 70 / 100;
```

```
Console.Write ("\n *** final Discount = "+ discount);
```

۴ با استفاده از مقادیر جدول زیر میزان تخفیف را محاسبه کنید، سپس برنامه را اجرا کرده، این مقادیر را به عنوان ورودی برنامه استفاده کنید.

قیمت کالا ۱	قیمت کالا ۲	قیمت کالا ۳	قیمت کالا ۴	میزان تخفیف	خروجی برنامه
۲۰۰	۳۰۰	۱۰۰	۵۰۰		

پودمان پنجم: حل مسائل شرطی

آیا خروجی برنامه با نتیجه‌ای که شما محاسبه کرده‌اید، یکسان است؟ نتیجه محاسبه اشتباه است. دلیل رخ دادن چنین خطایی چیست؟ به خطاهایی که در اثر اشتباه برنامه‌نویس در طراحی الگوریتم ایجاد می‌شود، خطاهای منطقی (Logic Error) می‌گویند. در صورت وجود این خطاها برنامه اجرا می‌شود ولی نتیجه اجرای برنامه، نتیجه مطلوب نیست.

۵ خطای منطقی برنامه را برطرف کنید.

۶ پس از برطرف کردن خطای منطقی، برنامه را با مقادیر جدول اجرا کنید.

قیمت کالا ۱	قیمت کالا ۲	قیمت کالا ۳	قیمت کالا ۴	میزان تخفیف	خروجی برنامه
۱۵۰	۱۰۰	۴۲۰	۲۳۰		

۷ برنامه را تغییر دهید تا علاوه بر میزان تخفیف، میزان مبلغی را که باید پردازیم، هم نمایش دهد.

۸ برنامه بالا را برای درصدهای متفاوت تخفیف مطابق جدول زیر تغییر دهید و برنامه را اجرا کنید.

تخفیف کالای اول	تخفیف کالای دوم	تخفیف کالای سوم	تخفیف کالای چهارم
۱۰٪	۱۵٪	۱۰٪	۶۵٪

۹ کد برنامه را طوری تغییر دهید تا میزان تخفیف کالاها را نیز از ورودی دریافت کند.

۱۰ در دستورات `Console.WriteLine` در کد برنامه از ابتدای رشته `\n` را برداشته، برنامه را اجرا کرده، خروجی برنامه را با خروجی مرحله قبل مقایسه کنید.

در زبان C# برای نشان دادن برخی نویسه‌ها و یا عملکردهای خاص در یک رشته از دنباله‌های فرار (Escape Sequences) استفاده می‌شود. دنباله فرار، یک دنباله معنی‌دار از نویسه‌هاست که با نویسه `\` شروع می‌شود. این دنباله در یک رشته معنای دیگری غیر از آنچه که در ظاهر دیده می‌شود پیدا می‌کند. در جدول ۴ برخی دنباله‌های معنی‌دار آمده است.

جدول ۴- عملکرد برخی دنباله‌های فرار

عملکرد دنباله	دنباله معنی‌دار
ایجاد یک بوق هشدار (Alarm)	<code>\a</code>
حذف یک کاراکتر (Backspace)	<code>\b</code>
بازگشت به ابتدای خط (Carriage Return)	<code>\r</code>
ایجاد یک خط خالی (New Line)	<code>\n</code>
ایجاد یک Tab فاصله افقی (Tab)	<code>\t</code>
ایجاد یک تک کوتیشن	<code>\'</code>
ایجاد یک دابل کوتیشن	<code>\"</code>
ایجاد یک <code>\</code> (Backslash)	<code>\\</code>



با کمک دنباله `\t` و ایجاد یک `Tab` فاصله می‌توان ستون‌های یک جدول متنی را زیر هم تراز کرد. جدول زیر را در خروجی یک برنامه ایجاد کنید.

```
Spring      Summer      Autumn      Winter
March       June         September   December
April       July         October     January
May         August      November    February
```

در مورد انواع خطاها در برنامه‌نویسی تحقیق کنید.

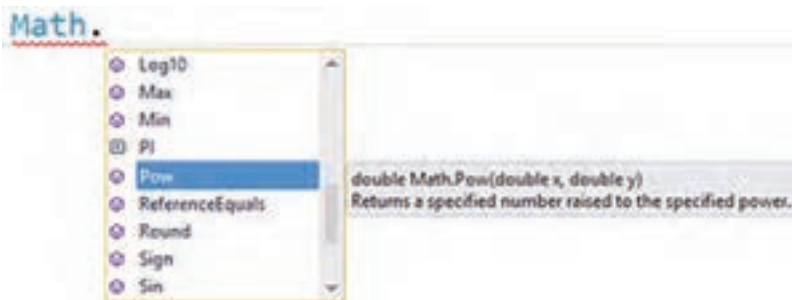


کارمند اداره‌ای هرماه ۵ درصد از حقوق خود را به سازمان حمایت از کودکان کار اهدا می‌کند. با فرض اینکه از حقوق این شخص هرماه ۱۰ درصد مالیات و ۷ درصد بیمه کسر شود، برنامه‌ای بنویسید که حقوق شخص را پس از کسر مالیات، بیمه و مبلغ اهدایی نمایش دهد.



کلاس Math

آیا قابلیت محاسبه عبارت‌های مثلثاتی با عملگرهای محاسباتی وجود دارد؟
 آیا با عملگرهای محاسباتی می‌توانید محیط میدان شهر ($2\pi r$) را محاسبه کنید؟
 آیا با عملگرهای محاسباتی می‌توانید حجم یک ظرف کره‌ای شکل ($\frac{4}{3}\pi r^3$) و جذر یک عدد (\sqrt{n}) را محاسبه کنید؟



شکل ۱- ویژگی‌ها و متدهای کلاس Math

برای محاسبه عبارت‌های ریاضی، نظیر عبارت‌های بالا از متدهای کلاس **Math** استفاده می‌شود. با نوشتن **Math** در سی‌شارپ و درج نقطه، فهرست متدهای این کلاس و عملکرد متد، نوع خروجی و ورودی‌های آن دیده می‌شود (شکل ۱).

عدد π که در ریاضیات کاربرد فراوان دارد، یکی از ویژگی‌های کلاس **Math** است و به صورت **Math.PI** در کدنویسی استفاده می‌شود.

با استفاده از راهنمای سی شارپ جدول ۵ را تکمیل کنید.



جدول ۵- برخی متدهای کلاس Math

نام متد	شرح متد	نوع داده ورودی	نوع داده خروجی	معادل ریاضی	معادل سی شارپ	پاسخ عبارت
Pow	مقدار عدد به توان را برمی گرداند.	double	double	2^5	Math.Pow(2,5)	
Sqrt	ریشه دوم(جذر) عدد را برمی گرداند.			$\sqrt{9}$	Math.Sqrt(9)	
Round	عدد را با توجه به تعداد اعشار گرد می کند.			گرد کردن عدد ۵/۶۱۹ تا رقم اعشار	Math.Round(5.619,2)	

با استفاده از متدهای کلاس Math معادل سی شارپ عبارات ریاضی زیر را بنویسید.

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$V = \frac{4}{3} \pi r^3$$

کارگاه ۳ به کارگیری متدهای کلاس Math

شهرداری قصد دارد در شهر یک بوستان دارای فواره آب و نگارخانه کوچک ایجاد کند. می خواهیم تعداد کاشی و سرامیک مورد نیاز برای کاشی کاری این فضاها را محاسبه کنیم.

1 پروژه های با نام MathProject ایجاد کنید.

2 قطعه کد زیر را برای محاسبه محیط حوض دایره ای فواره تکمیل کنید و در متد Main بنویسید.

```
//input radius of the circle
```

```
Console.WriteLine("Enter the radius in meter: ");
```

```
int radius = .....
```

```
//caculate circle environment
```

```
double env = 2 * Math.PI * radius;
```

```
//display circle environment
```

```
.....
```

۳ قطعه کدی بنویسید که با دریافت مساحت فضای مربع شکل نگارخانه، اندازه اضلاع آن را مشخص کند.

۴ با توجه به اضلاع نگارخانه مرحله ۳ دستوراتی به برنامه اضافه کنید که تعداد سرامیک با ابعاد ۳۰×۶۰ سانتی متر را برای فرش کردن نگارخانه محاسبه کند.

۵ با توجه به محیط حوض فواره به ارتفاع ۳۰ سانتی متر، دستوراتی به برنامه اضافه کنید که تعداد کاشی ۱۵×۱۵ سانتی متر برای پوشاندن دیواره حوض را نمایش دهد.

آنچه آموختیم:

برداشت



- ۱
- ۲
- ۳

جدول ارزشیابی شایستگی‌های غیر فنی، ایمنی و بهداشت و توجهات زیست محیطی

شایستگی‌ها	شرایط عملکرد (ابزار، مواد، تجهیزات، زمان، مکان و ...)	نتایج ممکن	استاندارد (شاخص‌ها/داوری/نمره‌دهی)	نمره
شایستگی‌های غیر فنی	استدلال، به‌کارگیری قوانین در فرایند کار - زبان فنی	قابل قبول	به‌کارگیری قوانین اولویت عملگرها برای تعیین نتیجه عبارت محاسباتی - بازگرداندن تنظیمات IDE به تنظیمات اولیه پس از استفاده از آن	۲
ایمنی و بهداشت	رعایت ارگونومی			
توجهات زیست محیطی		غیر قابل قبول	توجه به ایمنی و بهداشت محیط کارگاه	۱
نگرش	دقت در محاسبه نتیجه عبارت محاسباتی با توجه به اولویت انواع عملگرها			

نمره	استاندارد (شاخص‌ها/داوری/نمره‌دهی)	نتایج ممکن	شرایط عملکرد (ابزار، مواد، تجهیزات، زمان، مکان و ...)	مراحل کار
۳	نوشتن عبارت ریاضی در زبان برنامه‌نویسی - تغییر اولویت عملگرها - محاسبه نتیجه عبارت ریاضی - تعیین نوع متغیر برای ذخیره نتیجه عبارت - نوشتن عبارت ریاضی پیچیده و نیازمند به پرانتزهای متداخل و تعیین نتیجه عبارت	بالا تر از حد انتظار	مکان: کارگاه استاندارد رایانه تجهیزات: رایانه‌ای که نرم‌افزار IDE برنامه‌نویسی روی آن نصب باشد. زمان: ۱۰ دقیقه	نوشتن عبارات ریاضی به زبان برنامه‌نویسی
۲	نوشتن عبارت ریاضی در زبان برنامه‌نویسی - تغییر اولویت عملگرها - محاسبه نتیجه عبارت ریاضی	در حد انتظار		
۱	نوشتن عبارت ریاضی در زبان برنامه‌نویسی	پایین تر از حد انتظار		

عملگرهای انتساب

از جمله عملگرهای زبان‌های برنامه‌نویسی عملگر انتساب است. به عبارات زیر توجه کنید: در این عبارت، مقدار سمت راست (Y) در متغیر سمت چپ (X) قرار می‌گیرد. در این عبارت مراحل زیر به ترتیب اجرا می‌شود:

$$X = Y;$$

$$X = Y = Z = 0;$$

- ۱ متغیر Z با 0 مقداردهی می‌شود.
- ۲ متغیر Y با مقدار متغیر Z مقداردهی می‌شود.
- ۳ متغیر X با مقدار متغیر Y مقداردهی می‌شود.

عملگرهای دو عبارت $X=Y=Z$ و $X+Y+Z$ را اولویت بندی کرده، با هم مقایسه کنید.

فعالیت
کارگاهی





عملگرهای ریاضی "شرکت پذیری چپ" دارند در حالی که عملگرهای انتساب "شرکت پذیری راست" دارند.

علاوه بر عملگر انتساب = عملگرهای انتساب دیگری نیز وجود دارند. با توجه به مقادیر متغیرهای X و Y جدول ۶ را تکمیل کنید.

`int X = 10, Y = 3;`

جدول ۶- عملگرهای انتساب

نتیجه X	معادل	عبارت
13	$X = X + Y;$	$X += Y;$
	$X = X - Y;$	$X -= Y;$
	$X = X * Y;$	$X *= Y;$
	$X = X / Y;$	$X /= Y;$
	$X = X \% Y;$	$X \% = Y;$



اولویت عملگرهای انتساب، از عملگرهای ریاضی کمتر است.



به کمک هم گروه خود برنامه‌ای بنویسید که عددی از ورودی دریافت کرده، به ترتیب عملیات زیر را انجام دهد و خروجی نهایی را چاپ کند.

۱ عدد ورودی را در ۲ ضرب کند.

۲ حاصل را در ۵ ضرب کند.

۳ حاصل را با عدد ۱۰ جمع کند.

۴ حاصل را بر ۱۰ تقسیم کند.

۵ از حاصل، عدد یک را کم کند.

۶ حاصل را نمایش دهد.

۷ برنامه را با ورودی‌های مختلف اجرا کنید. خروجی برنامه با عدد ورودی چه ارتباطی دارد؟

عملگرهای افزایشی - کاهشی

به عملگرهای یکتایی ++ (plus plus) و -- (minus minus) که به ترتیب برای افزایش و کاهش مقدار یک متغیر به اندازه یک واحد به کار می‌روند، عملگرهای **افزایشی - کاهشی** می‌گوییم (جدول ۷).

جدول ۷- عملگرهای افزایشی - کاهشی

عبارت معادل	مثال	نام عملگر	عملگر
$X = X + 1$	X++ ++X	افزایشی	++
$X = X - 1$	--X X--	کاهشی	--

کارگاه ۴ عملگرهای افزایشی - کاهشی

برنامه‌ای بنویسید که تعداد هنرجویان دو کلاس ۱۰۱ و ۱۰۲ را دریافت کند و پس از انتقال یکی از هنرجویان از کلاس ۱۰۱ به کلاس ۱۰۲ تعداد هنرجویان دو کلاس را نمایش دهد.

۱ پروژه جدیدی با نام **Students Count** ایجاد کنید.

۲ دستورات زیر را در متد **Main** بنویسید.

```
int count101, count102;
```

```
Console.WriteLine("Enter count of student class 101:");
count101 = int.Parse(Console.ReadLine());
Console.WriteLine("Enter count of student class 102:");
count102 = int.Parse(Console.ReadLine());
Console.WriteLine("count of students class 101={0} ", count101--);
Console.WriteLine("count101={0} ", count101);
Console.WriteLine("count of students class 102={0} ", count102++);
Console.WriteLine("count102={0} ", count102);
```


۳ برنامه را اجرا کرده، خروجی برنامه را بنویسید.

آیا خروجی برنامه تعداد هنرجویان دو کلاس را به درستی نشان می‌دهد؟

۴ قسمت نمایش تعداد هنرجویان کلاس را مطابق کد زیر تغییر دهید و برنامه را دوباره اجرا کنید.

```
Console.WriteLine("count of students class 101={0} ", --count101);  
Console.WriteLine("count of students class 102={0} ", ++count102);
```

آیا خروجی برنامه تعداد هنرجویان دو کلاس را به درستی نشان می‌دهد؟

آیا مقدار متغیرها تغییر کرده است و به درستی نمایش داده می‌شود؟

عملگرهای ++ و -- عملگرهای یکتایی هستند و فقط بر روی متغیرها عمل می‌کنند.

یادداشت



۵ در برنامه به جای عملگرهای افزایشی - کاهشی، معادل آنها را قرار دهید و برنامه را اجرا کنید.

اگر عملگر افزایشی - کاهشی در یک عبارت یا دستور بعد از متغیر به کار رود، ابتدا متغیر با مقدار فعلی در محاسبات شرکت می‌کند؛ سپس مقدار متغیر یک واحد اضافه - کم می‌شود.

```
Console.WriteLine ("count of students class 101={0} ", count101--);
```

دستور بالا معادل دو دستور زیر است.

```
Console.WriteLine ("count of students class 101={0} ", count101);  
count101--;
```

اگر عملگر افزایشی - کاهشی در یک عبارت یا دستور قبل از متغیر به کار رود، ابتدا مقدار متغیر یک واحد اضافه - کم شده و متغیر با مقدار جدید در محاسبات بعدی شرکت می‌کند.

```
Console.WriteLine ("count of students class 101={0} ", --count101);
```

دستور بالا معادل دو دستور زیر است.

```
--count101;  
Console.WriteLine ("count of students class 101={0} ", count101);
```



- به کمک هم گروه خود معادل دستورات زیر را بنویسید.

```
Console.WriteLine("count of students class 102={0} ", count102++);
Console.WriteLine("count of students class 102={0} ", ++count102);
```

- با توجه به قطعه کد زیر و تغییرات متغیرهای X و Y جدول را تکمیل کنید.

```
int x = 12, y = -12;
Console.WriteLine ("x={0} y={1}", x++, ++y);
Console.WriteLine ("x={0} y={1}", --x, ++y);
y = ++x;
x = y--;
Console.WriteLine ("x={0} y={1}", x, y);
```

خروجی برنامه	y	x

عملگر الحاق دو رشته

به قطعه کد زیر توجه کنید. عملگر + علاوه بر اینکه یک عملگر محاسباتی است، عملگر رشته‌ای هم محسوب می‌شود. اگر حداقل یکی از عملوندهای عملگر + از نوع string باشد، این عملگر به عنوان عملگر الحاق دو رشته عمل می‌کند.

```
int a=5+7;  
string s= "ali"+"reza";  
string count="count of pencil : "+100;
```

در یک برنامه قطعه کد بالا را نوشته، محتوای متغیرهای a، s و count را نمایش دهید.

فعالیت
کارگاهی



نتیجه این دستورها چیست؟
به قطعه کد زیر توجه کنید.

```
string s = "ali";  
s+="reza";  
Console.WriteLine(s);
```

این قطعه کد را در متد Main بنویسید و برنامه را اجرا کنید. از خروجی برنامه چه نتیجه‌ای می‌گیرید؟

عملگر + برای الحاق دو رشته و عملگر += به عنوان عملگر انتساب الحاق دو رشته نیز به کار می‌رود.

یادداشت



پودمان پنجم: حل مسائل شرطی

فعالیت
کارگاهی



برنامه‌ای بنویسید که نام و نام خانوادگی را به صورت جداگانه از ورودی دریافت کند، سپس نام و نام خانوادگی را با استفاده از:
- عملگر الحاق دو رشته در یک متغیر ذخیره کرده، نمایش دهد.
- عملگر انتساب الحاق دو رشته در یک متغیر ذخیره کرده، نمایش دهد.

برداشت



آنچه آموختم:

- ۱.....
- ۲.....
- ۳.....

ارزشیابی مرحله ۲

نمره	استاندارد (شاخص‌ها/داوری/نمره دهی)	نتایج ممکن	شرایط عملکرد (ابزار، مواد، تجهیزات، زمان، مکان و ...)	مراحل کار
۳	استفاده از عملگرهای کاهش و افزایش و انتساب در برنامه - محاسبه نتیجه عبارت شامل عملگرهای کاهش و افزایش و انتساب - محاسبه نتیجه عبارت شامل + و += - انتخاب نوع متغیر برای ذخیره نتیجه عبارت	بالتر از حد انتظار	مکان: کارگاه استاندارد رایانه تجهیزات: رایانه‌ای که نرم‌افزار IDE برنامه‌نویسی روی آن نصب باشد زمان: ۱۰ دقیقه	به‌کارگیری عملگرهای افزایش و کاهش و انتساب
۲	استفاده از عملگرهای کاهش و افزایش و انتساب در برنامه - محاسبه نتیجه عبارت شامل عملگرهای کاهش و افزایش و انتساب	در حد انتظار		
۱	استفاده از عملگرهای کاهش و افزایش در برنامه	پایین‌تر از حد انتظار		

جدول ۸- عملگرهای مقایسه‌ای

عملکرد	معادل ریاضی	عملگر
برابری	=	==
نامساوی	≠	!=
کوچک‌تر	<	<
بزرگ‌تر	>	>
کوچک‌تر مساوی	≤	<=
بزرگ‌تر مساوی	≥	>=

رتبه اول کلاس شما چه کسی است؟ بیشترین نمره فعالیت کارگاهی را چه کسی کسب کرده است؟ میزان سرانه آب مصرفی در ایران در مقایسه با سایر کشورها چگونه است؟ در زندگی روزمره با سؤالات بسیار زیادی از این قبیل روبه‌رو هستیم که پاسخ به آنها نیاز به انجام عمل مقایسه دارد. در زبان‌های برنامه‌نویسی از عملگرهای مقایسه‌ای برای انجام عمل مقایسه استفاده می‌شود (جدول ۸).

به عبارتی که شامل عملگرهای مقایسه‌ای هستند یک عبارت منطقی می‌گویند که بیان‌کننده یک شرط است. جواب شرط یا نتیجه یک عبارت منطقی، در صورت درستی true و در صورت نادرستی false است. حاصل عبارت‌های منطقی را می‌توان در خروجی نمایش داد.

```
Console.WriteLine("flag = {0}", a>10);
```

نتیجه عبارت منطقی در چه نوع متغیری ذخیره می‌شود؟
متغیر flag از چه نوعی است؟ flag = a>10;

کنجکاوی



کارگاه ۵ عملگرهای مقایسه‌ای

در خبرنامه شرکت آب و فاضلاب کشور آمده است: متوسط سرانه مصرف آب در بخش خانگی در ایران به طور متوسط ۱۹۰ لیتر در روز به ازای هر نفر و در دنیا ۱۳۵ تا ۱۴۰ لیتر است. این خبر نشان می‌دهد مردم کشور ما نسبت به متوسط مردم دنیا مصرف بالاتری دارند. می‌خواهیم برنامه‌ای بنویسیم که سرانه مصرف آب در ایران را با سرانه مصرف آب در دنیا مقایسه کند.

- ۱ پروژه جدیدی با نام **WaterConsumption** ایجاد کنید.
- ۲ دستورات زیر را در متد **Main** بنویسید.

```
int iranWater = 190;
Console.WriteLine(" The average consumption in IRAN -> " + iranWater);
int worldWater = 135;
Console.WriteLine(" The average consumption in WORLD -> " + worldWater);
// آیا سرانه مصرف آب در ایران بیشتر است//
Console.WriteLine(" Is iranWater > worldWater: {0}", iranWater > worldWater );
```

- ۲ برنامه را اجرا کرده، خروجی آن را بنویسید.

کارگاه ۶ عملگرهای منطقی

می‌خواهیم برنامه‌ای بنویسیم که عدم قبولی یک هنرجو را با نمایش `false` به علت گرفتن نمره ۷ نشان دهد.

۱ پروژه‌ای با نام `Failure` ایجاد کنید.

۲ کد زیر را در متد `Main` بنویسید.

```
bool flag=false;
int score=7;
Console.WriteLine("score = {0}",score);
Console.WriteLine("is student passed:{0}, is score=7:{1}",flag==true,score==7);
```

۳ با هم گروهی خود در مورد خروجی این برنامه بحث کنید.

۴ برنامه را اجرا کنید.

عملگر `=` عمل انتساب را انجام می‌دهد. عبارت `a=b` برای مقداردهی متغیر `a` با مقدار متغیر `b` استفاده می‌شود، درحالی که عبارت `a==b` بیان می‌کند که "آیا `a` و `b` مساوی هستند؟" و نتیجه این عبارت `true` یا `false` است.

اولویت عملگرهای محاسباتی بالاتر از عملگرهای مقایسه‌ای است.

یادداشت



با توجه به مقادیر متغیرها جدول زیر را تکمیل کنید.

```
int a=5, failed=9, passed =15;
```

نتیجه	عبارت منطقی
	<code>passed >= 12</code>
	<code>failed + 2 < passed - a</code>
	<code>passed != 20</code>
	<code>failed == passed - a</code>

فعالیت کارگاهی



کارگاه ۷ اشکال زدایی عبارات (Debugging)

می‌خواهیم عددی را دریافت کرده، محاسبات زیر را انجام دهیم.

- افزایش یک واحد به ورودی
 - ۵ برابر کردن نتیجه قبل
 - کاهش یک واحد از نتیجه قبل
 - نصف کردن نتیجه آخر
- ۱ پروژه‌های با نام **Debug1** ایجاد کنید.
 - ۲ کد زیر را در متد **Main** بنویسید.

```
int num1;  
Console.WriteLine("Enter num1 : ");  
num1=int.Parse(Console.ReadLine());  
num1++;  
num1*5;  
--num1;  
num1/2;  
Console.WriteLine("num1={0} ", num1);  
Console.ReadKey();
```

۲ برنامه را با استفاده از کلید **F6** ترجمه کنید.

۴ خطاهای این برنامه و علت رخ دادن آنها را در جدولی که مشابه پنجره لیست خطا در **VS** طراحی شده، بنویسید.

شماره ردیف	خطا	علت

کارگاه ۸ اشکال زدایی در رشته‌ها

- ۱ پروژه‌های با نام **Debug2** ایجاد کنید.
- ۲ کد زیر را در متد **Main** بنویسید.

```
bool result = true;  
Console.WriteLine(result > 0);  
string s1 = "Ali";  
string s2 = "Reza";  
s1 += s2;  
s1 = s1 - s2;  
float f = 100 - s1;  
s1 = -s1;
```

پودمان پنجم: حل مسائل شرطی

۳ برنامه را با استفاده از کلید F6 ترجمه کنید.

۴ خطاهای این برنامه و علت رخ دادن آنها را در جدول زیر بنویسید.

شماره ردیف	خطا	شماره خط	شماره ستون	علت

عملوند عملگرهای >= و < و < نمی‌توانند از نوع string و bool باشند.

یادداشت



کارگاه ۹ اشکال زدایی خطای تقسیم بر صفر

فرد خیری تعدادی دفترچه به یک مؤسسه خیریه اهدا کرده است. می‌خواهیم برنامه‌ای بنویسیم که تعداد دفترچه‌ها و تعداد بچه‌های این مؤسسه را از ورودی دریافت کرده، تعیین کند به هریک از بچه‌ها چند دفترچه تعلق می‌گیرد.

۱ پروژه‌ای با نام Debug3 ایجاد کنید.

۲ کد زیر را در متد Main بنویسید.

```
int booklet, children;  
Console.WriteLine("Enter number of booklet : ");  
booklet = int.Parse(Console.ReadLine());  
Console.WriteLine("Enter number of children : ");  
children = int.Parse(Console.ReadLine());  
int count = booklet / children;  
Console.WriteLine("count={0} ", count);  
Console.ReadKey();
```

۳ برنامه را اجرا کنید.

تعداد دفترچه‌ها و بچه‌های مؤسسه را مطابق جدول زیر وارد کرده، جدول را تکمیل کنید.

خروجی	تعداد بچه‌ها	تعداد دفترچه‌ها
	۲۰۰	۵۰۰
	۰	۲۰۰

۴ خطاهای این برنامه و علت رخ دادن آن را در کادر زیر بنویسید.
تفاوت خطای این کارگاه با خطای کارگاه ۷ و ۸ چیست؟

خطا	علت

برداشت



آنچه آموختم:

- ۱.....
- ۲.....
- ۳.....

ارزشیابی مرحله ۳

مرحله کار	شرایط عملکرد (ابزار، مواد، تجهیزات، زمان، مکان و ...)	نتایج ممکن	استاندارد (شاخص ها/داوری/نمره دهی)	نمره
استفاده از عملگرهای مقایسه‌ای	مکان: کارگاه استاندارد رایانه تجهیزات: رایانه‌ای که نرم‌افزار IDE برنامه‌نویسی روی آن نصب باشد. زمان: ۱۰ دقیقه	بالاتر از انتظار	به‌کارگیری عملگرهای مقایسه‌ای و تعیین نتیجه عبارت منطقی - تشخیص و رفع خطای استفاده نادرست از عملگرهای ریاضی، مقایسه‌ای و انتساب	۳
		در حد انتظار	به‌کارگیری عملگرهای مقایسه‌ای و تعیین نتیجه عبارت منطقی	۲
		پایین تر از حد انتظار	به‌کارگیری عملگرهای مقایسه‌ای	۱

عملگرهای منطقی

آیا در بین دوستان شما افراد کمتر از ۱۵ سال و بزرگ‌تر از ۲۰ سال وجود دارد؟ آیا معدل سال گذشته شما بین ۱۲ و ۱۷ است؟ آیا شرایط لازم برای حضور هنرجویان در اردو فراهم است؟
در مسائلی از این قبیل با عملیات منطقی روبه‌رو هستیم و تصمیم نهایی ما به ترکیب شرطها بستگی دارد.
برای ترکیب عملگرهای مقایسه‌ای می‌توان از عملگرهای منطقی استفاده کرد. عبارتی که شامل عملگرهای منطقی باشد یک عبارت منطقی است.

با هم گروهی خود پویانمایی عملگرهای منطقی را مشاهده کنید و خلاصه‌ای از برداشت خود را در سه خط بنویسید.

۱
۲
۳

فعالیت
کارگاهی



کارگاه ۱۰ عملگر منطقی &&

جدول ۹- عملگر منطقی &&

p	q	p && q
false	false	false
false	true	false
true	false	false
true	true	true

عملگر منطقی && دو عملوند دارد. در جدول ۹ نتیجه اجرای عملگر && براساس حالت‌های مختلف عملوندهای آن آمده است.

رمز قفل یک چمدان مسافرتی 25 است. می‌خواهیم برنامه‌ای بنویسیم که باز شدن قفل را شبیه‌سازی کند. ارقام رمز را جداگانه از ورودی دریافت می‌کنیم.

۱ پروژه‌ای با نام **Lock** ایجاد کنید.

۲ قطعه کد زیر را در متد **Main** وارد کنید.

```
byte x, y;
y = byte.Parse(Console.ReadLine());
x = byte.Parse(Console.ReadLine());
Console.WriteLine((x==2) && (y==5));
```



در صورتی که خروجی true باشد، قفل باز می‌شود. ۳ برنامه را با مقادیر متفاوت برای x و y اجرا کرده، جدول را تکمیل کنید.

خروجی	y	x
	5	3
	4	2
true		
false		



- عبارت منطقی بنویسید که عدد ورودی که شماره یک ماه سال است را بررسی کند؟
- حاصل عبارت $X < 14 \ \&\& \ X > 10$ در چه صورتی true است؟



عملوندهای یک عملگر منطقی می تواند متغیری از نوع bool، یک شرط و یا مقادیر true و یا false باشد.



جدول زیر را تکمیل کنید.

نتیجه	مقدار	اولویت بندی	عبارت منطقی	شرح
	score=13	score >=0 && score <=20	score >=0 && score <=20	نمره هنرجو بین ۰ تا ۲۰ است
	score=26			
		① ③ ②		شرط ثابت نام در همایش نمره بالای ۱۲ و معدل بالای ۱۴ است
				هوا ابری نباشد و نمره هنرجو بالای ۱۵ باشد



اولویت عملگرهای مقایسه‌ای بالاتر از عملگرهای منطقی است.

کارگاه ۱۱ عملگر منطقی ||

قفل چمدان مسافرتی کارگاه ۱۰ خراب شده است و در صورت درست بودن حداقل یکی از ارقام رمز هم باز می شود. می خواهیم برنامه‌ای بنویسیم که باز شدن قفل را در این وضعیت شبیه‌سازی کند. ارقام رمز را جداگانه از ورودی دریافت می کنیم.

پودمان پنجم: حل مسائل شرطی

عملگر منطقی `||` دو عملوند دارد. در جدول ۱۰ نتیجه اجرای عملگر `||` براساس حالت‌های مختلف عملوندهای آن آمده است.

جدول ۱۰- عملگر منطقی `||`

p	q	p q
false	false	false
false	true	true
true	false	true
true	true	true

- ۱ پروژه‌ای با نام `BadLock` ایجاد کنید توجه داشته باشید که رمز قفل همان 25 است.
- ۲ قطعه کد زیر را در متد `Main` بنویسید.

```
byte x, y;
y = byte.Parse(Console.ReadLine());
x = byte.Parse(Console.ReadLine());
Console.WriteLine((x==2) || (y==5));
```

- ۳ برنامه را با مقادیر متفاوت برای `x` و `y` اجرا کرده، جدول را تکمیل کنید.

خروجی	y	x
	4	2
	4	3
true		
false		

- عبارت منطقی بنویسید که نشان دهد عدد ورودی در محدوده مجاز برای نمره هنرجو نیست.
 - حاصل عبارت `X > 14 || X < 10` در چه صورتی `true` است؟





جدول زیر را تکمیل کنید.

نتیجه	مقدار	اولویت بندی	عبارت منطقی	شرح
	months=5	① ③ ② months < 1 months > 31	months < 1 months > 31	عدد ورودی در محدوده شماره روزهای یک ماه نیست.
	months=40			
				شرط ثبت نام در همایش معدل بالای ۱۷ یا قبولی در آزمون ورودی
				هوا ابری باشد یا نمره هنرجو کمتر از ۱۵ باشد

کارگاه ۱۲ عملگر منطقی !

مدیریت هنرستان تصمیم گرفته است اگر هوا مناسب باشد، هنرجویان را به اردو ببرد. می‌خواهیم برنامه‌ای بنویسیم که وضعیت هوا را از ورودی دریافت کند و وضعیت رفتن به اردو را تعیین کند.

- ۱ پروژه‌ای با نام WeatherCamp ایجاد کنید.
- ۲ قطعه کد زیر را در متد Main بنویسید.

string weather;

Console.WriteLine(" weather:");

weather = Console.ReadLine();

Console.WriteLine("weather is not cloudy: {0}", weather!="cloudy");

۳ برنامه را اجرا کرده، جدول زیر را تکمیل کنید.

خروجی	وضعیت هوا
	cloudy
	sunny

پودمان پنجم: حل مسائل شرطی

عملگر منطقی ! یک عملوند دارد. در جدول ۱۱ نتیجه اجرای عملگر ! براساس حالت‌های مختلف عملوند آن آمده است.

جدول ۱۱- عملگر منطقی!

p	! p
false	true
true	false

جدول زیر را تکمیل کنید.

فعالیت
کارگاهی



نتیجه	مقدار	اولویت بندی	عبارت منطقی	شرح
	no=2	④ ① ③ ② !(no >= 1 && no <=4)	!(no >= 1 && no <=4)	شماره فصل نیست
				هوا ابری است یا هنرجو نمره قبولی نگرفته است. (بین ۱۲ تا ۲۰)
				معادله درجه ۲ ریشه مضاعف دارد یا ریشه ندارد.

یادداشت



اولویت عملگرها در یک عبارت به ترتیب عبارت است از: محاسباتی، مقایسه‌ای و منطقی برنامه با استفاده از نتایج عملگرهای مقایسه‌ای و منطقی، برای انجام عملیات بعدی تصمیم‌گیری می‌کند.

کنجکاوی



کدام یک از عملگرهای منطقی یکتایی و کدام دوتایی هستند؟
 یکتایی
 دوتایی



- در برنامه‌ای سن و اندازه قد دو نفر دریافت می‌شود. برای هر کدام از مسائل زیر شرط مناسب بنویسید.
 الف) آیا این دو نفر هم سن و هم قد هستند؟
 ب) آیا این دو نفر هم سن یا هم قد هستند؟
 - از کارمندی که بالای یک میلیون حقوق می‌گیرد و کمتر از ۳ فرزند دارد، مالیات کسر می‌شود.
 برنامه‌ای بنویسید که با دریافت حقوق و تعداد فرزندان با نمایش `true` یا `false` نشان دهد که آیا از این کارمند مالیات کسر می‌شود؟
 - اگر کاربری با نام کاربری "admin" و گذرواژه "Pass123" داشته باشیم. برنامه‌ای بنویسید که نام کاربری و گذرواژه را دریافت کند و صحت آنها را با نمایش `true` یا `false` نمایش دهد.

۴ خروجی قطعه برنامه‌های زیر را بررسی کنید.

الف)

```
Console.WriteLine("Enter num1:");
int num1 = int.Parse(Console.ReadLine());
Console.WriteLine("Enter num2:");
int num2 = int.Parse(Console.ReadLine());
Console.WriteLine(" {0} greater than {1} -> {2}", num1, num2, num1 > num2);
Console.WriteLine(" {0} less than {1} -> {2}", num1, num2, num1 < num2);
Console.WriteLine(" {0} equal to {1} -> {2}", num1, num2, num1 == num2);
Console.WriteLine(" {0} not equal to {1} -> {2}", num1, num2, num1 != num2);
```

ب)

```
Console.WriteLine("Enter an integer:");
int myInt = int.Parse(Console.ReadLine());
bool isLessThan10 = myInt < 10;
bool isBetween0And5 = (0 <= myInt) && (myInt <= 5);
Console.WriteLine("Integer less than 10? {0}", isLessThan10);
Console.WriteLine("Integer between 0 and 5? {0}", isBetween0And5);
```

پودمان پنجم: حل مسائل شرطی

برداشت



آنچه آموختم:

- ۱.....
- ۲.....
- ۳.....

ارزشیابی مرحله ۴

مراحل کار	شرایط عملکرد (ابزار، مواد، تجهیزات، زمان، مکان و ...)	نتایج ممکن	استاندارد (شاخص‌ها/داوری/نمره‌دهی)	نمره
استفاده از عملگرهای منطقی	مکان: کارگاه استاندارد رایانه تجهیزات: رایانه‌ای که نرم‌افزار IDE برنامه‌نویسی روی آن نصب باشد. زمان: ۱۰ دقیقه	بالاتر از حد انتظار	استفاده از عملگرهای منطقی در ترکیب شرطها و تعیین نتیجه آنها - انتخاب نوع متغیر برای خروجی عبارت منطقی - محاسبه نتیجه عبارت منطقی ترکیبی پیچیده و طولانی	۳
		در حد انتظار	استفاده از عملگرهای منطقی در ترکیب شرطها و تعیین نتیجه آنها	۲
		پایین تر از حد انتظار	استفاده از عملگرهای منطقی در ترکیب شرطها	۱
<p>معیار شایستگی انجام کار :</p> <p>کسب حداقل نمره ۲ از مرحله استفاده از نوشتن عبارات ریاضی به زبان برنامه‌نویسی</p> <p>کسب حداقل نمره ۲ از بخش شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش</p> <p>کسب حداقل میانگین ۲ از مراحل کار</p>				

جدول ارزشیابی پایانی

شرح کار:

- ۱ نوشتن عبارات ریاضی به زبان برنامه‌نویسی
- ۲ به کارگیری عملگرهای افزایشی و کاهشی و انتساب
- ۳ استفاده از عملگرهای مقایسه‌ای
- ۴ استفاده از عملگرهای منطقی

استاندارد عملکرد:

با استفاده از دانش ریاضی و شناخت انواع عملگرها و اولویت آنها نسبت به یکدیگر در زبان برنامه‌نویسی، عبارات محاسباتی ایجاد و نتیجه آنها را محاسبه کرده و خطاهای احتمالی آن را برطرف کند.

شاخص‌ها:

شماره مرحله کار	شاخص‌های مرحله کار
۱	تبدیل عبارات ریاضی ساده و پیچیده به زبان برنامه‌نویسی
۲	به کارگیری عملگرهای کاهشی یا افزایشی در عبارت در صورت نیاز - محاسبه نتیجه عبارت شامل عملگر + یا =
۳	به کارگیری عملگرهای ریاضی و مقایسه‌ای و انتساب مورد نیاز در یک عبارت با توجه به اولویت آنها - رفع خطاهای احتمالی عبارت منطقی
۴	به کارگیری عملگر منطقی برای ترکیب عبارات منطقی در صورت نیاز - محاسبه نتیجه عبارت ترکیبی منطقی - مقداردهی متغیر با عبارات محاسباتی و منطقی و رفع خطاهای احتمالی آن

شرایط انجام کار و ابزار و تجهیزات:

مکان: کارگاه رایانه مطابق استاندارد تجهیزات هنرستان‌ها
 تجهیزات: رایانه‌ای که نرم‌افزار IDE برنامه‌نویسی روی آن نصب باشد.
 زمان: ۴۰ دقیقه (نوشتن عبارات ریاضی به زبان برنامه‌نویسی ۱۰ دقیقه - به کارگیری عملگرهای افزایشی و کاهشی و انتساب ۱۰ دقیقه - استفاده از عملگرهای مقایسه‌ای ۱۰ دقیقه - استفاده از عملگرهای منطقی ۱۰ دقیقه)

معیار شایستگی:

ردیف	مرحله کار	حداقل نمره قبولی از ۳	نمره هنرجو
۱	نوشتن عبارات ریاضی به زبان برنامه‌نویسی	۲	
۲	به کارگیری عملگرهای افزایشی و کاهشی و انتساب	۱	
۳	استفاده از عملگرهای مقایسه‌ای	۱	
۴	استفاده از عملگرهای منطقی	۱	
	شایستگی‌های غیرفنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش: استدلال - زبان فنی رعایت ارگونومی دقت در محاسبه نتیجه عبارت محاسباتی ترکیبی با توجه به اولویت انواع عملگرها	۲	
	میانگین نمرات		*

* حداقل میانگین نمرات هنرجو برای قبولی و کسب شایستگی، ۲ است.

واحد یادگیری ۱۰

شایستگی کار با ساختار شرطی

آیا تا به حال پی برده اید

- در زبان سی شارپ چند نوع ساختار شرطی وجود دارد؟
- چگونه می توان ساختار `if - else` ساده را کدنویسی کرد؟
- چگونه می توان `if`های تودرتو را با دستورات کمتری بازنویسی کرد؟
- چگونه می توان برخی ساختارهای شرطی `if` را با `switch` بازنویسی کرد؟

هدف از این شایستگی تشخیص نیاز به استفاده از ساختار شرط در برنامه و به کارگیری ساختارهای شرطی و کنترل خطاهای آن است

استاندارد عملکرد

با استفاده از دانش ساختارهای شرطی و نوع داده شمارشی، چگونگی عملکرد ساختارهای شرطی در الگوریتم برنامه را تحلیل و آن را با استفاده از انواع داده شمارشی و دستورات شرطی بنویسید.



از دوران کودکی تاکنون سؤالات زیادی داشتیم که با **اگر** شروع می‌شد و مسیر زندگی انسان را مشخص می‌کرد.

اگر دوچرخه داشتیم، چه می‌شد؟

اگر دیپلم گرفته بودم؛

اگر دانشجو بودم؛

بعضی از این **اگرها** دور از دسترس و برخی دیگر دست‌یافتنی است و هنگامی که اتفاق می‌افتد؛ اکثر انسان‌ها در تعقیب **اگرهای** دیگرند.

مثلاً **اگر** دوچرخه داشتیم، چه می‌شد؟ **اگر** دوچرخه داشتیم می‌گفتم **اگر** می‌توانستم برای دوچرخه‌ام زنگ بخرم چه می‌شد؟ بعد از آن در این فکر بودم که **اگر** در تیم دوچرخه سواری منطقه عضو می‌شدم و در اینجا می‌خواهیم برای **اگرها** برنامه بنویسیم.

به کمک هم‌گروهی خود **اگرهای** کتاب دانش فنی پایه را در جدول زیر وارد کنید.

فعالیت
گروهی



ردیف	عنوان	شرط	درست بودن شرط	نادرست بودن شرط
۱	تفکیک زباله	اگر زباله خشک باشد	در سطل مخصوص زباله خشک قرار بده	—
۲	خرید کالا			خرید نکردن
۳	بازی سنگ، کاغذ، قیچی			
۴				
۵				

پودمان پنجم: حل مسائل شرطی

در برخی از مسائل اگرها ساده نیست و شامل چندین شرط است. در این بخش می‌خواهیم یکی از اگرهای اولیه کودکی را تبدیل به یک قطعه کد کنیم. «اگر دوچرخه داشتم بازی می‌کردم».

```
if (x=="bicycle")  
    Console.WriteLine("Playing");
```

در قطعه کد بالا متغیر X نشان دهنده نوع وسیله‌ای است که داریم. با همین اگر ساده توانستیم یک قطعه کد کوچک بنویسیم. همان‌طور که مشاهده می‌کنید واژه اگر در زبان برنامه نویسی به if تبدیل شد.

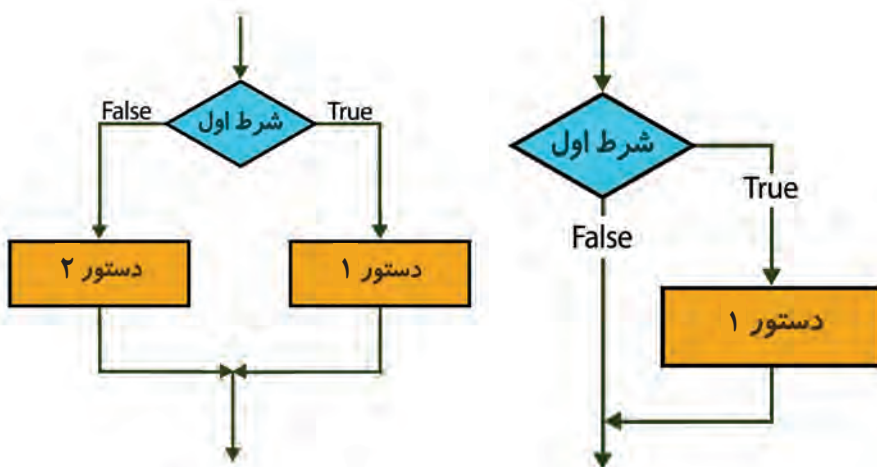
در قطعه کد بالا نوع متغیر X چیست؟

کنجکاوی



دو روندنمای زیر را با هم مقایسه کرده و تعیین کنید کدام مناسب مسئله بالا است.

فعالیت
کارگاهی



شکل ۲- ساختار دستور if

شکل کلی if ساده

(عبارت شرطی) if
دستور;

کارگاه ۱ دستور شرطی ساده

می‌خواهیم برنامه‌ای بنویسیم که مزایای دو روش پرداخت قبوض آب و برق را مشخص کند:

✓ مراجعه به بانک: پرداخت قبوض

✓ استفاده از پرداخت الکترونیکی: پرداخت قبوض، کاهش ترافیک

۱ پروژه‌ای با نام EPayment ایجاد کنید.

۲ کد زیر را در متد Main وارد کنید.

```
Console.WriteLine("Enter your choice // 1 or 2");
byte x = byte.Parse(Console.ReadLine());
if (x==1)
    Console.WriteLine("Pay bills");
if (x==2)
{
    Console.WriteLine("Pay bills");
    Console.WriteLine("reduction of traffic ");
}
```

۳ برنامه را اجرا کرده و نتایج آن را مشاهده کنید.

۴ بعد از پرانتز **if** دوم یک؛ قرار داده، نتیجه را با نتیجه مرحله قبل مقایسه کنید.

۵ بلاک شرط دوم را بردارید و برنامه را اجرا کنید و نتیجه را با نتیجه مرحله ۳ مقایسه کنید.

مجموعه دستوراتی را که داخل {} قرار می‌گیرد. یک بلاک می‌گویند.

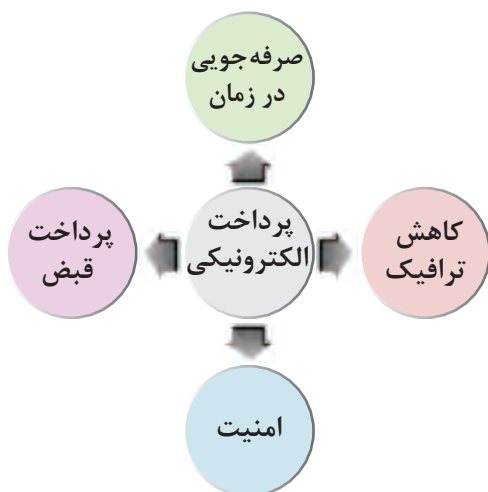
یادداشت



فعالیت کارگاهی



با توجه به شکل مزایای دیگر پرداخت الکترونیکی قبوض را به کد اضافه کنید.



کارگاه ۲ ساختار شرطی if _ else

- می‌خواهیم قسمتی از برنامه هدایت یک خودروی بدون سرنشین را برای عبور از چراغ قرمز طراحی کنیم.
- ۱ پروژه جدیدی با نام **VCondition** ایجاد کنید.
 - ۲ برنامه زیر را کامل کرده، در متد **Main** بنویسید.

```
if ( light=="red")
    vehicle Condition = "stop";
else
    vehicle Condition = "move";
Console.WriteLine(Vehicle Condition );
```

۳ کدام یک از روندنماهای شکل ۱ مناسب این برنامه است؟

۴ روندنمای این برنامه را رسم کنید.

۵ برنامه را به صورت زیر تغییر دهید.

اگر چراغ قرمز بود، ابتدا ماشین ترمز (break) کرده، سپس متوقف شود؛ در غیراین صورت با زدن یک بوق حرکت کند.

برنامه‌ای بنویسید که میزان سوخت موجود در یک خودرو را از ورودی خوانده، اگر کمتر از ۵ لیتر بود پیام مناسب با رنگ قرمز و در غیراین صورت پیام مناسب دیگری با رنگ سبز نمایش دهد.

فعالیت
کارگاهی



کارگاه ۳ بلاک در ساختار شرطی if

می‌خواهیم برنامه‌ای طراحی کنیم تا با دریافت یک رشته برای زوج یا فرد بودن روزهای هفته، برنامه بعدازظهرهای علی را تعیین کند.

روز	۱۵ تا ۱۶:۳۰	۱۷ تا ۱۸:۳۰	۱۸:۴۰ تا ۱۹	۱۹:۱۵ تا ۲۰:۴۵
روزهای شنبه - دوشنبه - چهارشنبه	انجام تکالیف روزانه	باشگاه ورزشی	فریضه نماز	کلاس # C
روزهای یکشنبه - سه شنبه - پنج شنبه	انجام تکالیف روزانه	کلاس زبان انگلیسی	فریضه نماز	سرزدن به پدربزرگ

۱ پروژه جدیدی با نام Week ایجاد کنید.

۲ کد زیر را در متد Main بنویسید.

آیا برنامه خطا دارد؟

```
string day = Console.ReadLine();
if (day == "odd">//condition true
{
    Console.WriteLine("Studying");
    Console.WriteLine("English class");
    Console.WriteLine("Say praying");
    Console.WriteLine("Visit Grand");
}
if (day == "Even") //condition true
    Console.WriteLine("Studying");
    Console.WriteLine("gym");
    Console.WriteLine("Say praying");
    Console.WriteLine("c# programming Class ");
```

۳ خروجی برنامه در صورتی که ورودی «even» باشد، چیست؟

۴ خروجی برنامه در صورتی که ورودی «odd» باشد، چیست؟

دلیل اشتراک پاسخ در مراحل ۳ و ۴ چیست؟

۵ خطای برنامه را رفع کنید.

اگر خروجی برنامه نادرست است با توجه به آنچه قبلاً آموختید نوع خطا را مشخص کنید و مشکل برنامه را برطرف کنید.

آیا می توان برنامه را با ساختار if - else نوشت؟

برنامه‌ای بنویسید که مدت زمان تقریبی انتظار در صف بانک را با شرایط زیر به دست آورد. برنامه باید شماره آخرین نفر ورودی به بانک، شماره آخرین نفری که در حال انجام کار است و تعداد بجه‌های فعال را دریافت کند. تعداد بجه‌های فعال، باید بیشتر از صفر باشد. فرض کنید میانگین مدت زمان لازم برای انجام کار هر نفر در بجه معادل ۲ دقیقه باشد.

فعالیت منزل



کارگاه ۴ اجرای گام به گام if _ else

در یک اداره ۲۴ طبقه، ۲ دستگاه آسانسور، مخصوص طبقات زوج و فرد و یک نفر جهت راهنمایی مراجعان وجود دارد که با پرسش از افراد، آسانسور مناسب را مشخص می‌کند. می‌خواهیم با هدف مکانیزه کردن قسمتی از تشکیلات ساختمانی از یک برنامه کمک بگیریم تا با وارد کردن شماره طبقه، آسانسور مناسب را مشخص کند. آسانسور A مخصوص طبقات فرد و آسانسور B مخصوص طبقات زوج است.

۱ پروژه جدیدی با نام **ChElevator** ایجاد کنید.

۲ کد زیر را در **Main** وارد کنید.

```
Console.WriteLine("Enter the floor");
int floor = int.Parse(Console.ReadLine());
if (floor % 2 == 0)
    Console.WriteLine("Elevator A");
else
    Console.WriteLine("Elevator B");
```

۳ اگر مراجعه‌کننده به‌طور سهوی عدد ۳۴ را وارد کند خروجی برنامه چیست؟

۴ برنامه را طوری تغییر دهید تا در صورتی که کاربر عددی خارج از محدوده تعداد طبقات وارد

کرد پیام مناسبی چاپ شود.

۵ خروجی برنامه را با مقادیر روبه‌رو **Trace** کنید.

خروجی	floor
	10
	25
	-6
	5.2
	floor13

کارگاه ۵ عملگر سه تایی

می‌خواهیم برنامه‌ای بنویسیم تا با دریافت سن فرد تعیین کند آیا صلاحیت دریافت گواهی نامه رانندگی دارد؟

۱ پروژه جدیدی با نام **DLicence** ایجاد کنید.

اگر سن فرد مساوی یا بزرگ‌تر از ۱۸ سال باشد می‌تواند در آزمون رانندگی شرکت کند و در غیراین صورت این امکان وجود ندارد. در نتیجه باید با استفاده از یک دستور شرطی، سن بررسی شود.

۲ کد زیر را تکمیل کنید و در متد **Main** وارد کنید.

```
if (age >= 18)
    status = "Allowed";
else
    status = "Not Allowed";
Console.WriteLine(status);
```


۳ خروجی برنامه را با مقادیر زیر Trace کنید.

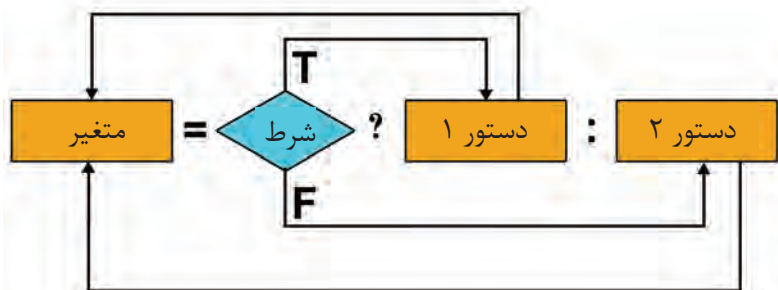
خروجی	age
	20
	18
	10

۴ متغیرهای age و status را اعلان کنید.

۵ به جای کد قبلی، کد زیر را وارد کنید.

```
Console.WriteLine("How old are you?");
int age = int.Parse(Console.ReadLine());
string s;
s = (age >= 18 ? "Allowed": " Not Allowed");
Console.WriteLine (s);
```

۶ خروجی برنامه را در دو حالت مقایسه کنید.



شکل ۳- عملگر سه تایی

در برخی موارد که دستور شرطی، از یک دستور، برای بخش if و else استفاده می کند؛ می توان از عملگر سه تایی (ternary) به صورت زیر استفاده کرد. هنگامی که شرط درست باشد خروجی این عملگر حاصل دستور 1 و هنگامی که شرط نادرست باشد، حاصل دستور 2 است (شکل ۳).

شکل کلی عملگر سه تایی

دستور 2 : دستور 1 ? (عبارت شرطی) = خروجی

۷ برنامه را با ورودی twenty اجرا کنید . چه خطاهای رخ می دهد؟ چرا؟

۸ مدیریت خطا انجام دهید.

پودمان پنجم: حل مسائل شرطی

دستور زیر را جایگزین خط دوم کد مرحله ۵ کنید.

```
bool result = in.TryParse (Console.ReadLine () , out int age);
```

برنامه را با ورودی twenty اجرا کنید. آیا خروجی برنامه درست است؟
متد TryParse همانند متد Parse رشته ورودی را به عدد تبدیل می کند ولی موفقیت یا عدم موفقیت عمل تبدیل را در خروجی از نوع داده منطقی (bool) تعیین می کند.
در کد بالا در صورت وارد کردن عدد، رشته به عدد تبدیل شده در متغیر age (ورودی دوم متد TryParse) قرار داده می شود و متغیر result با true مقداردهی می شود ولی اگر ورودی عدد نباشد متغیر age با صفر و متغیر result با false مقداردهی می شوند.

با استفاده از متد Tryparse از بروز خطا در برنامه به خاطر ورود اطلاعات نادرست جلوگیری کردیم ولی خروجی برنامه درست نیست.
۹ برنامه را اصلاح کنید.

```
Console.WriteLine (" How old are you");
```

```
bool result = int.TryParse (Console.ReadLine() , out int age);
```

```
if (result)
```

```
{  
    string s = (age = 18 ? "Allowed" : "Not Allowed");  
    Console.WriteLine (s);  
}
```

برنامه را با ورودی twenty اجرا کنید. نتیجه را با مرحله قبل مقایسه کنید.
برنامه را تکمیل کنید تا در صورت وارد کردن مقادیر غیر عددی پیام مناسب نمایش داده شود.

دو مورد از کارگاه‌های قبل که قابلیت استفاده از عملگر سه تایی را دارند، بازنویسی کنید.

فعالیت
کارگاهی



برداشت



آنچه آموختم:

- ۱.....
- ۲.....
- ۳.....

جدول ارزشیابی شایستگی‌های غیرفنی، ایمنی و بهداشت و توجهات زیست‌محیطی

شایستگی‌ها	شرایط عملکرد (ابزار، مواد، تجهیزات، زمان، مکان و ...)	نتایج ممکن	استاندارد (شاخص‌ها/داوری/نمره دهی)	نمره
شایستگی‌های غیرفنی	استدلال، تجزیه و تحلیل منطق‌های قوانین/ مفاهیم - زبان فنی	قابل قبول	ارائه دلیل برای انتخاب عملگرهای منطقی ترکیب شرط‌ها و دستورات شرطی مورد استفاده در برنامه	۲
ایمنی و بهداشت	رعایت ارگونومی			
توجهات زیست‌محیطی		غیرقابل قبول	توجهات به ایمنی و بهداشت محیط کارگاه	۱
نگرش	دقت در تشخیص ساختار شرطی مناسب هنگام تبدیل الگوریتم به برنامه در مسئله مطرح شده			

ارزشیابی مرحله ۱

مراحل کار	شرایط عملکرد (ابزار، مواد، تجهیزات، زمان، مکان و ...)	نتایج ممکن	استاندارد (شاخص‌ها/داوری/نمره دهی)	نمره
به‌کارگیری دستور شرطی در برنامه	مکان: کارگاه استاندارد رایانه تجهیزات: رایانه‌ای که نرم‌افزار IDE برنامه‌نویسی روی آن نصب باشد. زمان: ۲۰ دقیقه	بالاتر از حد انتظار	رسم روندنمای برنامه با دستورات شرطی و نوشتن کد آن - استفاده از عملگر سه‌تایی شرطی - مدیریت و رفع خطاهای برنامه	۳
		در حد انتظار	رسم روندنمای برنامه با دستورات شرطی و نوشتن کد آن - استفاده از عملگر سه‌تایی شرطی	۲
		پایین‌تر از حد انتظار	رسم روندنمای برنامه با دستورات شرطی	۱

کارگاه ۶ عبارت شرطی تودرتو

می‌خواهیم برنامه‌ای بنویسیم که کاربر معتبر برای ورود به سیستم را مشخص کند.

نام کاربری معتبر: reza

گذرواژه صحیح: 123456

۱ پروژه جدیدی با نام `UserPass` ایجاد کنید.

۲ کد زیر را در متد `Main` وارد کنید.

```
Console.WriteLine("Enter your user name");
string user = Console.ReadLine();
Console.WriteLine("Enter your password");
string pass = Console.ReadLine();
if (user == "reza")
    if (pass == "123456")
        Console.WriteLine("Welcome " + user);
    else
        Console.WriteLine("The password is incorrect");
else
    Console.WriteLine("Unkown user");
Console.ReadKey();
```

۳ با توجه به برنامه جدول زیر را تکمیل کنید.

شرط	دستور
اگر نام کاربری و گذرواژه صحیح است؛	
اگر نام کاربری صحیح است و گذرواژه صحیح نیست.	
اگر نام کاربری صحیح نیست؛	

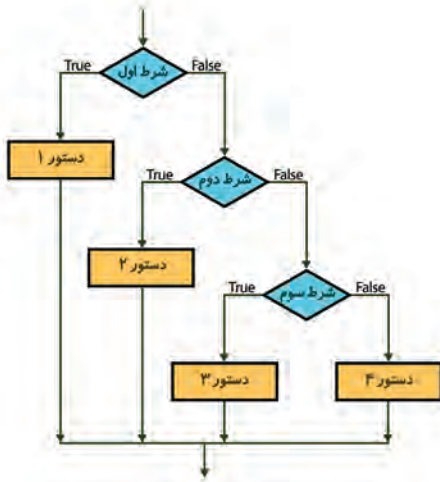


- چرا هیچ کدام از **if** ها بلاک {} ندارند؟
- هر کدام از **else** ها مربوط به کدام **if** است؟ به کمک هنرآموز خود روش کلی برای این مسئله ارائه دهید.



شکل ۴- ساختارهای شرطی در یک نگاه

برای تعیین محدوده عبارت شرطی که بیش از یک دستور دارد، استفاده از بلاک {} ضروری است.



شکل ۵- ساختار شرطی تودرتو

با کمک هم گروهی خود، برای روندنمای زیر یک مسئله طراحی کنید. می توانید از موارد شکل ۵ ایده بگیرید.

- ✓ داشتن شرایط مختلف برای استخدام در شرکت برنامه نویسی
- ✓ عیب یابی کامپیوتر
- ✓ ثبت نام دانشگاه
- ✓ کنترل کیفیت یک محصول غذایی



اگر درون یک ساختار شرطی، ساختار شرطی دیگری باشد؛ به آن عبارات شرطی تودرتو (Nested conditional statements) می گویند.






پودمان پنجم: حل مسائل شرطی

عبارات شرطی تو در تو و عملگرهای منطقی

عبارات شرطی تو در تو را می توان با استفاده از عملگرهای منطقی ساده تر و خواناتر نوشت. اما به کار بردن آنها دقت زیادی لازم دارد.

کارگاه ۷ عبارات شرطی با عملگرهای منطقی

می خواهیم با استفاده از نرم افزار پیش بینی هوا امکان بازدید هنرجویان از نمایشگاه کتاب را در فصل بهار بررسی کنیم.

امکان برگزاری	هوا	دما	ردیف
عالی		بیشتر از ۱۹ درجه	۱
خوب		بیشتر از ۱۹ درجه	۲
مناسب		کمتر از ۱۹ درجه	۳
غیر ممکن		کمتر از ۱۹ درجه	۴

- ۱ پروژه جدیدی با نام **Weather** ایجاد کنید.
- ۲ متغیرهای مورد نیاز برنامه را اعلان کنید.

```
Console.WriteLine("Enter the air temperature");  
int temp = int.Parse(Console.ReadLine());  
Console.WriteLine("Enter the weather conditions: sunny /cloudy / partly cloudy/ rainy");  
string condition = Console.ReadLine();
```

۳ برای ردیف ۱ جدول، قطعه کد زیر را وارد کنید.

```
if (temp > 19)
  if (condition == "sunny")
    Console.WriteLine("Excellent");
```

۴ برنامه را اجرا کرده، ورودی‌هایی به برنامه بدهید که پیام **Excellent** را نمایش دهد.

۵ کد زیر را جایگزین خطوط بالا کنید.

```
if ((temp > 19) && (condition == "sunny"))
  Console.WriteLine("Excellent");
```

۶ برنامه را دوباره اجرا کرده، نتایج را مقایسه کنید.

۷ کد ردیف ۲ جدول را با استفاده از عملگرهای منطقی بنویسید.

۸ کد زیر را برای ردیف ۳ جدول به برنامه اضافه کرده، سپس برنامه را اجرا و نتیجه را یادداشت کنید.

```
if (temp < 19)
{
  if (condition == "sunny")
    Console.WriteLine("suitable");
  if (condition == "partly cloudy ")
    Console.WriteLine("suitable");
}
```

۹ به کمک هنرآموز خود کد این مرحله را با استفاده از عملگرهای منطقی بازنویسی کرده، سپس نتیجه را با کد قبلی مقایسه کنید.

۱۰ کد ردیف ۴ جدول را با عملگرهای منطقی به برنامه اضافه کنید.

برنامه را طوری بازنویسی کنید که اگر کاربر در ردیف ۳ و ۴ دمای کمتر از ۸ درجه وارد کرد؛ برگزاری اردو غیرممکن شود.

فعالیت
کارگاهی



- ۱ پروژۀ جدیدی با نام **Sort** ایجاد کنید.
- ۲ برنامه زیر را تکمیل کرده، در متد **Main** وارد کنید.

```
if ((a > b) && (a > c))
    if (b > c)
        Console.WriteLine("{0} {1} {2}", a, b, c);
    else
        Console.WriteLine("{0} {1} {2}", a, c, b);

else if ((b > a) && (b > c))
    if (a > c)
        Console.WriteLine("{0} {1} {2}", b, a, c);
    else
        Console.WriteLine("{0} {1} {2}", b, c, a);

else if ((c > a) && (c > b))
    if (a > b)
        Console.WriteLine("{0} {1} {2}", c, a, b);
    else
        Console.WriteLine("{0} {1} {2}", c, b, a);
```

- ۳ نوع مناسب برای متغیرهای **a**، **b**، **c** چیست؟
- ۴ برنامه را طوری تغییر دهید تا صحت داده‌ها نیز بررسی شود. صحت داده‌ها: سن نباید منفی و از ۱۱۰ بیشتر باشد.

به کمک هم گروهی خود سعی کنید برنامه را با طرح و نقشه دیگری بنویسید.

فعالیت
گروهی



فعالیت منزل



- برنامه‌ای بنویسید که در انتخابات شورای دانش‌آموزی مدرسه، نفر برتر را از بین ۴ نفر شرکت‌کننده در انتخابات تعیین کند. برنامه باید با دریافت ۴ عدد که مشخص‌کننده تعداد آرای هر نفر است، نفری که بیشترین رأی را به خود اختصاص داده معین کند.

- برنامه‌ای بنویسید که نفر برتر مسابقات دوی ۱۰۰ متر کلاس را تعیین کند. در این برنامه باید با دریافت زمان ۴ شرکت‌کننده برحسب ثانیه، نفر اول مشخص شود.

کارگاه ۹ ترکیب عبارات منطقی، محاسباتی و شرطی

می‌خواهیم برنامه‌ای بنویسیم که با دریافت سه عدد a ، b ، c مشخص کند آیا این سه عدد می‌توانند اضلاع یک مثلث باشند؟

روش اول: برای حل مسئله، از قضیهٔ همار در هندسه استفاده می‌کنیم. این قضیه بیان می‌کند که در هر مثلث مجموع هر دو ضلع از ضلع دیگر بزرگ‌تر است^۱. برای مثال سعی کنید مثلثی با ابعاد ۱ و ۲ و ۲۰ ترسیم کنید. چنین مثلثی وجود ندارد. زیرا هرگز دو ضلع ۱ و ۲ به هم نخواهند رسید. مطابق قضیه همار، مجموع دو ضلع باید از ضلع سوم بزرگ‌تر باشد که در این مثال $۲+۱ > ۲۰$ نیست.



زمانی که داده‌های ورودی از سه شرط $(a + b > c, a + c > b, b + c > a)$ به‌درستی عبور کنند، به خروجی **true** می‌رسیم و اگر در هر نقطه از مسیر منحرف شوند؛ خروجی **false** خواهد بود. برای این کار به متغیری (count) نیاز داریم که در مسیر حرکت به سوی مقصد، تعداد شرط‌هایی را که به‌درستی از آن عبور می‌کنیم؛ بشمارد و هر وقت تعداد شمارش شده به عدد ۳ رسید؛ خروجی **true** شود.

۱ پروژه جدیدی با نام **Triangle** ایجاد کنید.

۲ برنامه زیر را تکمیل کرده و در متد **Main** بنویسید.

```
int count = 0;
if (b + c > a) count++;
if (a + c > b) count++;
if (a + b > c) count++;
if (count == 3)
    Console.WriteLine(true);
else
    Console.WriteLine(false);
```

۱- به‌بیان‌دیگر کوتاه‌ترین مسیر بین دو نقطه یک خط مستقیم است.

پودمان پنجم: حل مسائل شرطی

```
if (a + b <= c)
    Console.WriteLine(false);
else if (a + c <= b)
    Console.WriteLine(false);
else if (b + c <= a)
    Console.WriteLine(false);
else
    Console.WriteLine(true);
```

۳ برنامه را با ۳ مقدار ورودی اجرا کنید.

روش دوم: می‌توانیم از شرط تودرتو استفاده کنیم.

در این حالت true را در چهارمین دستور قرار می‌دهیم، زیرا تنها زمانی دستور چهارم اجرا می‌شود که دستورهای یک تا سه اجرا نشده باشند. در این ساختار شرطی، دیگر نیازی به شمارنده نیست.

۴ برنامه را اجرا کنید و نتایج را با کد قبلی مقایسه کنید.

روش سوم: در این روش از عملگرهای منطقی استفاده می‌کنیم.

```
Console.WriteLine(a + b > c && a + c > b && b + c > a);
```

اگر به عملگر && دقت کنید؛ متوجه می‌شوید که خروجی این عملگر زمانی true است که همه عملوندهای آن true باشد.

۵ برنامه را اجرا کرده، نتایج را دوباره با مراحل قبل مقایسه کنید.

- الگوریتم و روندنمای ۳ روش بالا را کشیده و با هم مقایسه کنید. به نظر شما کدام روش بهتر است؟ چرا؟
- روش سوم را با استفاده از عملگر || بنویسید (از هنرآموز خود کمک بگیرید).

فعالیت کارگاهی



با توجه به کارگاه ۱۰ صفحه ۲۳۷ برنامه‌ای بنویسید که با دریافت ۲ عدد x و y به عنوان رمز قفل شکلی مشابه زیر ترسیم کند.

فعالیت منزل



برداشت



آنچه آموختم:

- ۱
- ۲
- ۳

مرحله کار	شرایط عملکرد (ابزار، مواد، تجهیزات، زمان، مکان و ...)	نتایج ممکن	استاندارد (شاخص ها/داوری/نمره دهی)	نمره
به کارگیری دستورات شرطی متداخل در برنامه	مکان: کارگاه استاندارد رایانه تجهیزات: رایانه‌ای که نرم افزار IDE برنامه نویسی روی آن نصب باشد. زمان: ۲۰ دقیقه	بالاتر از حد انتظار	رسم روندنما برای برنامه با دستورات شرطی متداخل و نوشتن کد - به کارگیری عملگرهای منطقی برای ترکیب شرطها در صورت نیاز - رفع خطاهای برنامه	۳
		در حد انتظار	رسم روندنما برای برنامه با دستورات شرطی متداخل و نوشتن کد - به کارگیری عملگرهای منطقی برای ترکیب شرطها در صورت نیاز	۲
		پایین تر از حد انتظار	رسم روندنما برای برنامه با دستورات شرطی متداخل	۱

کارگاه ۱۰ | تبدیل if به Switch

می خواهیم برنامه‌ای بنویسیم که با دریافت یک فصل، ماه‌های آن را تعیین کند.

۱ پروژه جدیدی با نام **Season** ایجاد کنید.

۲ قطعه کد زیر را تکمیل کرده، در متد **Main** بنویسید.

```
if (season=="Bahar")
    Console.WriteLine("Farvardin, Ordibehesht, Khordad");
else if (season=="Tabestan")
    Console.WriteLine("Tir, Mordad, Shahrivar");
else if (season=="Paeiz")
    Console.WriteLine("Mehr, Aban, Azar");
else if (season=="Zemestan")
    Console.WriteLine("Dey, Bahman, Esfand");
else
    Console.WriteLine("Season Wrong");
```

۳ برنامه را اجرا و خروجی را مشاهده کنید.

پودمان پنجم: حل مسائل شرطی

برنامه ساده است اما نکته قابل توجه در این برنامه بررسی عبارت season در چندین حالت است. در زبان سی شارپ می توان برای بررسی حالت های مختلف یک عبارت از دستور switch استفاده کرد. ساختار این دستور به صورت زیر است :

شکل کلی دستور switch

<pre>switch(عبارت) { case مقدار ۱: دستور ۱; break; case مقدار ۲: دستور ۲; break; case مقدار ۳: دستور ۳; break; default: دستور ۴; break; }</pre>	<p>اگر حاصل یا مقدار عبارت برابر مقدار ۱ بود دستور ۱ انجام شده و با دستور break از ساختار خارج می شود.</p> <p>اگر برابر مقدار ۲ بود دستور ۲ انجام شده، با دستور break از ساختار خارج می شود (برای بقیه مقادیر نیز همین روش اجرا می شود).</p> <p>در نهایت، در صورتی که حاصل برابر هیچ یک از مقادیر نباشد دستور ۴ (default) انجام می شود و با دستور break از ساختار خارج می شود.</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

برنامه زیر را جایگزین برنامه قبلی کرده، برخی دستورات ناقص را تکمیل کنید.

```
string season ;
Console.WriteLine("Enter Season:");
season = Console.ReadLine();
switch (season)
{
  case "Bahar":
    Console.WriteLine("Farvardin, Ordibehesht, Khordad");
    break;
  case "Tabestan":
    Console.WriteLine("Tir, Mordad, Shahrivar");
    break;
  case "Paeiz ":
    _____
    _____
  case "Zemestan":
    _____
    break;
  default:
    Console.WriteLine("Season Wrong");
    _____
}
```

۵ برنامه را اجرا و نتایج را با کد قبلی مقایسه کنید.

در دستور **switch** برای قرار دادن مجموعه‌ای از دستورات در یک **case** یا **default**، نیازی به بلاک نیست.

یادداشت



شماره کارگاه	رقم سمت راست
5	1
3	2
11	3
10	4
4	5
2	6
8	7
6	8
7	9
9	0

کارگاه ۱۱ ساختار switch

می‌خواهیم برنامه‌ای بنویسیم که با دریافت شماره دانش‌آموزی و تعیین رقم سمت راست آن، شماره یک کارگاه از کارگاه‌های قبلی را برای ارائه در کلاس به او اختصاص دهد. برای جدا کردن رقم سمت راست یک عدد صحیح می‌توانیم باقی‌مانده تقسیم عدد بر ۱۰ را محاسبه کنیم.

تفاوت عدد و رقم چیست؟

کنجکاوی



۱ پروژه جدیدی با نام **IdSudent** ایجاد کنید.

۲ کد زیر را در متد **Main** وارد کنید. حالت‌های ۳ تا ۸ را مطابق جدول تکلیف، کدنویسی کنید.

```
long x = long.Parse(Console.ReadLine());
x = x % 10;
switch (x)
{
case 1:
    Console.WriteLine("5");
    break;
case 2:
    Console.WriteLine("3");
    break;
case 9:
    Console.WriteLine("7");
    break;
case 0:
    Console.WriteLine("9");
    break;
}
```

۳ در برنامه بالا **default** استفاده نشده است، آیا برنامه خطا دارد؟

۴ برنامه قبل را با ساختار **if** بازنویسی کنید و هر دو برنامه را با هم مقایسه کنید.

پودمان پنجم: حل مسائل شرطی

۵ مدیریت خطا انجام دهید.

هنگام اجرا به جای عدد ورودی، یک رشته وارد کنید. خطاهای حاصل را ترجمه کنید. با تغییر کد از خطا جلوگیری کنید.

۶ برنامه را اجرا و شماره تمرین خود را یادداشت کنید.

شماره تمرین خود را در کلاس اعلام کرده، هم گروهی احتمالی خود را پیدا کنید. گروه‌ها باید برنامه اختصاصی خود را در کلاس با شیوه‌ای خلاقانه شرح دهند.

فیلم

فیلم شماره ۱۰۲۲۱: اشکال زدایی switch



فیلم را مشاهده کرده و برداشت را تکمیل کنید.

برداشت

آنچه آموختم:



۱.....

۲.....

۳.....

خواص	میوه	ردیف
ضد سرطان، خون‌ساز، درمان ضعف		۱
تصفیه‌کننده خون، ضد سرطان		۲
تب‌بر، ضد نرمی استخوان		۳
کاهش کلسترول خون، کمک به سلامت لثه		۴
تنظیم فشار خون، مقوی مغز، شادابی پوست		۵
کاهش وزن، بهبود سردرد و معده درد		۶

برنامه‌ای بنویسید که با انتخاب نام میوه‌های جدول روبه‌رو خواص آنها را نمایش دهد. اگر میوه در فهرست نبود پیام مناسبی نمایش دهد.

فعالیت منزل



کارگاه ۱۲ | ایجاد فهرست در کنسول

r	صفحه نمایش قرمز شود.
g	صفحه نمایش سبز شود
b	صفحه نمایش آبی شود
w	صفحه نمایش سفید شود
y	صفحه نمایش زرد شود

شکل ۶- فهرست رنگ‌ها بر اساس انتخاب نویسه

می‌خواهیم برنامه‌ای بنویسیم که یک فهرست رنگ تولید کند، تا با انتخاب هر یک از نویسه‌های زیر به وسیله کاربر، صفحه نمایش تغییر رنگ دهد.

۱ پروژۀ جدیدی به نام **MenuColor** ایجاد کنید.

۲ برنامه زیر را در متد **Main** وارد کنید و بخش‌های ناقص برنامه را تکمیل کنید.

```
Console.WriteLine("What's your favorite color?");
Console.WriteLine("[r] red");
Console.WriteLine("[g] green");
Console.WriteLine("[b] blue");
Console.WriteLine("[w] white");
Console.WriteLine("[y] yellow");
Console.Write(" Enter your choice: ");
```

```
char choice = char.Parse(Console.ReadLine());
```

```
switch (choice)
```

```
{
```

```
case 'r':
```

```
    Console.BackgroundColor = ConsoleColor.Red;
```

```
    Console.Clear();
```

```
    break;
```

```
case 'g':
```

```
    _____
```

```
    _____
```

```
    _____
```

```
case 'b':
```

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

```
    Console.Clear();
```

```
    break;
```

```
default:
```

```
    _____
```

```
    _____
```

```
    _____
```

```
}
```

پودمان پنجم: حل مسائل شرطی

- ۳ برای نویسه g و w و y برنامه را تکمیل کنید.
- ۴ با ورودی‌های مختلف برنامه را اجرا کنید.
- ۵ یک نویسه خارج از فهرست تعیین شده به عنوان ورودی بدهید. صفحه نمایش چه رنگی می‌شود؟
- ۶ کد برنامه را طوری تکمیل کنید که فهرست رنگ مانند شکل ۵ در صفحه نمایش دیده شود.
- ۷ به فهرست برنامه امکان انتخاب رنگ قلم صفحه نمایش را هم اضافه کنید.

کارگاه ۱۳ | جزئیات دستور switch و شرط منطقی

می‌خواهیم برنامه‌ای بنویسیم که با دریافت نام یک دستگاه کامپیوتری نوع ورودی، خروجی و یا ورودی و خروجی بودن آنها را اعلام کند.

- ۱ پروژه جدیدی با نام **IODevice** ایجاد کنید.
- ۲ برنامه زیر را در متد **Main** وارد کنید.

```
Console.WriteLine("enter your device");
string x = Console.ReadLine();
switch (x)
{
    case "keyborad":
    case "Mouse":
    case "Scanner":
        Console.WriteLine("{0} is input device ",x);
        break;
    case "Monitor":
    case "Printer":
    case "Headphone":
        Console.WriteLine("{0} is output device ",x);
        break;
    case "Modem":
    case "Network card":
    case "Audio Card":
        Console.WriteLine("{0} is Both Input-OutPut Devices ", x);
        break;
    default:
        Console.WriteLine("unknown");
        break;
}
```

- ۳ برنامه را اجرا کنید.
- ۴ قابلیت شناسایی دستگاه‌های بیشتری را برای برنامه فراهم کنید.



یک ساختار `switch` کوچک می‌تواند به ازای دو یا چند مقدار یک مجموعه عملیات واحد را انجام دهد و این حالت معادل عملگر منطقی «یا» بین شرط هاست.

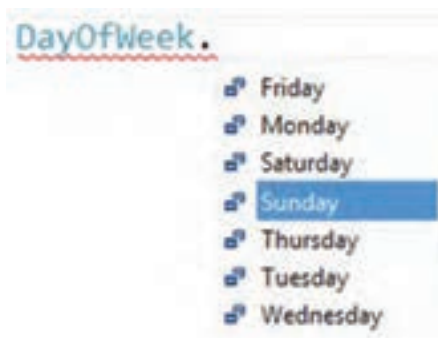
انواع داده شمارشی (Enumeration)

می‌خواهیم برنامه‌ای بنویسیم که روزهای هفته را براساس شماره روز نگهداری کند؛ بنابراین لازم است تعداد هفت ثابت به صورت زیر تعریف کنیم:

```
const int Sunday = 0;
const int Monday = 1;
const int Tuesday = 2;
const int Wednesday = 3;
const int Thursday = 4;
const int Friday = 5;
const int Saturday = 6;
```

آیا در این روش امکان انجام عملیات محاسباتی روی ثابت‌ها برای به دست آوردن نام دیگر وجود دارد؟ برای مثال می‌توان به یک ثابت دو واحد اضافه کرد و نام دو روز بعد را به دست آورد؟ برای بالا بردن خوانایی می‌توانیم از نوع داده شمارشی استفاده کنیم. نوع داده شمارشی مجموعه‌ای از چند نام دلخواه است که اجازه می‌دهد به جای اعداد ثابت و بی‌معنی، از واژه‌های با معنی استفاده کنیم. در زبان سی شارپ انواع داده شمارشی آماده بسیاری تعریف شده است. در جدول زیر اسامی برخی از داده‌های شمارشی را می‌بینید:

داده شمارشی	شرح
DayOfWeek	روزهای هفته
ConsoleColor	۱۶ رنگ اصلی



نوع داده شمارشی `DayOfWeek` شامل اسامی هفت روز هفته با شماره ۰ تا ۶ است.

شکل کلی دسترسی به اعضای نوع داده شمارشی

نام عضو. نوع داده شمارشی

```
DayOfWeek.Monday
```

به کد تغییر رنگ زمینه کنسول توجه کنید:

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

اعلان متغیری از نوع داده شمارشی را به صورت زیر انجام می‌دهیم.

نام متغیر نوع داده شمارشی

```
DayOfWeek      day ;
```

متغیر day از نوع داده شمارشی `DayOfWeek` است و می‌تواند با یکی از اعضای این نوع داده شمارشی به صورت زیر مقداردهی شود.

مقدار = نام متغیر

```
day = DayOfWeek.Friday;
```

کارگاه ۱۴ داده شمارشی رنگ‌ها

می‌خواهیم برنامه‌ای بنویسیم که اسامی ۱۶ رنگ اصلی را نمایش دهد.

۱ پروژه جدیدی به نام `Colors` ایجاد کنید.

۲ دستورات زیر را در متد `Main` بنویسید.

```
ConsoleColor c = ConsoleColor.Blue;
```

```
Console.BackgroundColor = c + 3;
```

```
Console.Clear();
```

۳ برنامه را اجرا کرده، خروجی را مشاهده کنید.

۴ دستورات زیر را به مرحله ۲ اضافه کنید.

```
c += 1;
```

```
Console.ForegroundColor = c;
```

```
Console.WriteLine (c);
```

۵ دستوراتی اضافه کنید که نام هر ۱۶ رنگ را با همان رنگ زمینه نمایش دهد.

۶ کد زیر را بنویسید تا شماره رنگ `Green` در نوع داده شمارشی `ConsoleColor` را نمایش دهد.

```
Console.WriteLine ((int) ConsoleColor.Green);
```

در نوع داده شمارشی برای دسترسی به عدد صحیح هر یک از عضوی آن، از تبدیل صریح برای تبدیل نوع داده شمارشی به عدد به صورت زیر استفاده می‌کنیم:

نام عضو. نوع داده شمارشی (`int`)

برنامه‌ای بنویسید که شماره تمام رنگ‌های نوع داده شمارشی `ConsoleColor` را با همان رنگ نمایش دهد.

فعالیت
کارگاهی





با استفاده از نوع داده شمارشی `DayOfWeek` برنامه‌ای بنویسید که اسامی روزهای هفته را همراه با شماره روز نمایش دهد.

تعریف نوع داده شمارشی

در زبان سی شارپ، برنامه‌نویس می‌تواند نوع داده شمارشی جدیدی به وسیله کلید واژه `enum` به صورت زیر تعریف کند.

نام دلخواه `enum`

```
}
```

فهرستی از نام‌ها

```
{
```

محل تعریف نوع داده شمارشی معمولاً خارج از کلاس و در فضای نام است. روش نام‌گذاری نوع داده شمارشی روش پاسکال است. در فهرست نام‌ها، هر نام با علامت کاما (,) از نام دیگر جدا می‌شود. در مثال زیر نوع داده شمارشی `MontOfYear` برای نام ماه‌های سال میلادی به صورت خلاصه تعریف شده است:

`enum MontOfYear`

```
{
```

```
Jan, Feb, Mar, Apr, May, Jun, Aug, Sep, Oct, Nov, Dec
```

```
}
```

هریک از اعضای نوع داده شمارشی معادل یک عدد صحیح ثابت است. این اعداد به طور پیش فرض از عدد صفر شروع شده و به ترتیب یک واحد اضافه می‌شوند. البته می‌توان عدد دیگری را به نام‌ها اختصاص داد. در مثال زیر نوع داده شمارشی `Fasl` برای اسامی فصل‌های سال تعریف شده است.

`enum Fasl`

```
{
```

```
Bahar = 1,
```

```
Tabestan,
```

```
Paez,
```

```
Zemestan
```

```
}
```

در این تعریف `Bahar` معادل با عدد یک و `Tabestan` برابر دو است.

در نوع داده شمارشی `Fasl` نام `Zemestan` معادل با چه عددی است؟



می‌خواهیم برنامه‌ای بنویسیم که شماره یک فصل را از ورودی دریافت کرده، اسامی ماه‌های آن را نمایش دهد. **۱** پروژه جدیدی به نام **Fasl** ایجاد کنید.

۲ نوع داده شمارشی **Season** را خارج از کلاس **Program** و در فضای نام اعلان کنید. برای افزایش خوانایی برنامه، پیشنهاد می‌شود هنگام اعلان نوع داده شمارشی، مقدار عددی هر عضو نوشته شود.

`enum Season`

```
{
    Winter = 1,
    Spring = 2,
    Summer = 3,
    Fall = 4
}
```

۳ دستورات زیر را برای دریافت شماره فصل و تبدیل شماره به نام فصل، در متد **Main** بنویسید.

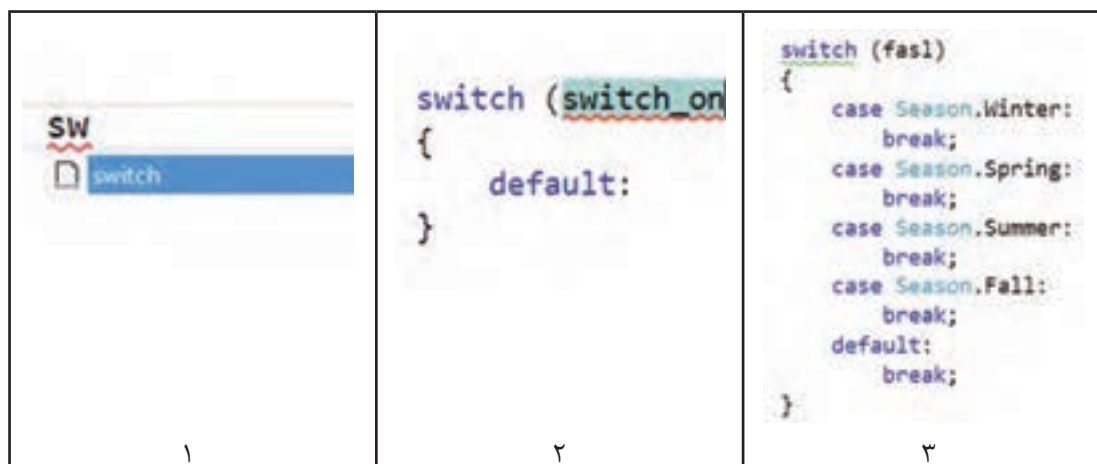
`Console.WriteLine ("Enter the Season number (1-4): ");`

`int no = int.Parse (Console.ReadLine ());`

`Season fasl = (Season) no;`

۴ دستور **switch** را به همراه تمامی **case**‌های آن بنویسید.

برای نوشتن دستور **switch** به کمک کدهای کوتاه (Code Snippets) عبارت **sw** را نوشته (۱) و دوبار کلید **Tab** را بزنید (۲). به جای عبارت **switch-on** داخل پرانتز، نام متغیر شمارشی یعنی **fasl** نوشته و دوبار **Enter** را بزنید (۳).



۵ برای بررسی صحت ورودی دستورات مناسب به برنامه اضافه کنید.

اگر شماره وارده در محدوده مجاز نباشد، پیام مناسب نمایش داده شود.



- کدام یک از دستورات مقاردهی قطعه کد زیر نادرست است؟ خطای آنها چیست؟

DayOfWeek day;

day = Monday;

day = DayOfWeek.Wednesday;

Console.WriteLine (day);

day = 5;

day = (DayOfWeek) 0;

Console.WriteLine (day);

Console.WriteLine ((int) day);

- دستوراتی که خطا دارد را حذف کرده، کد را در متد Main نوشته و اجرا کنید.



برای میزان تحصیلات ولی هنرجو نوع داده شمارشی تعریف کنید. برنامه‌ای بنویسید که شماره معادل مدارک تحصیلی را نمایش دهد.

ارزشیابی مرحله ۳

نمره	استاندارد (شاخص‌ها/داوری/نمره‌دهی)	نتایج ممکن	شرایط عملکرد (ابزار، مواد، تجهیزات، زمان، مکان و ...)	مراحل کار
۳	تعیین ساختار شرطی مناسب برای برنامه- رسم روندنمای برنامه و نوشتن کد-دسترسی به اعضای نوع داده شمارشی - اعلان متغیر نوع داده شمارشی و مقداردهی آن - تعریف نوع داده شمارشی - تبدیل نوع عددی به نوع داده شمارشی - رفع خطاهای برنامه	بالتر از حد انتظار	مکان: کارگاه استاندارد رایانه تجهیزات: رایانه‌ای که نرم‌افزار IDE برنامه‌نویسی روی آن نصب باشد. زمان: ۲۵ دقیقه	توسعه برنامه
۲	تعیین ساختار شرطی مناسب برای برنامه- رسم روندنمای برنامه و نوشتن کد-دسترسی به اعضای نوع داده شمارشی - اعلان متغیر نوع داده شمارشی و مقداردهی آن - تعریف نوع داده شمارشی	در حد انتظار		
۱	تعیین ساختار شرطی مناسب برای برنامه - رسم روندنمای برنامه - دسترسی به اعضای نوع داده شمارشی	پایین تر از حد انتظار		

معیار شایستگی انجام کار :

کسب حداقل نمره ۲ از مرحله به‌کارگیری دستور شرطی در برنامه

کسب حداقل نمره ۲ از بخش شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش

کسب حداقل میانگین ۲ از مراحل کار

شرح کار:			
۱ به کارگیری دستور شرطی در برنامه	۲ به کارگیری دستورات شرطی متداخل در برنامه		
۳ توسعه برنامه			
<p>استاندارد عملکرد:</p> <p>با استفاده از دانش ساختارهای شرطی و نوع داده شمارشی، چگونگی عملکرد ساختارهای شرطی در الگوریتم برنامه را تحلیل و آن را با استفاده از انواع داده شمارشی و دستورات شرطی بنویسد.</p>			
شاخص‌ها			
شماره مرحله کار	شاخص‌های مرحله کار		
۱	ارائه راه‌حل مسئله شرطی یا رسم روندنمای راه‌حل - نوشتن برنامه روندنما - مدیریت و رفع خطاهای برنامه		
۲	ارائه راه‌حل مسئله شرطی متداخل یا رسم روندنمای راه‌حل - نوشتن برنامه روندنما - به کارگیری عملگرهای منطقی برای ترکیب شرطها در صورت نیاز - رفع خطاهای احتمالی ساختارهای شرطی پیچیده		
۳	استفاده از دستورات شرطی جایگزین برای ساده‌سازی برنامه - تعریف نوع داده شمارشی - به کارگیری نوع داده شمارشی در برنامه و رفع خطاهای آن		
<p>شرایط انجام کار و ابزار و تجهیزات:</p> <p>مکان: کارگاه رایانه مطابق استاندارد تجهیزات هنرستان‌ها</p> <p>تجهیزات: رایانه‌ای که نرم‌افزار IDE برنامه‌نویسی روی آن نصب باشد.</p> <p>زمان: ۶۰ دقیقه (به کارگیری دستور شرطی در برنامه ۲۰ دقیقه - به کارگیری دستورات شرطی متداخل در برنامه ۲۰ دقیقه - توسعه برنامه ۲۵ دقیقه)</p> <p>معیار شایستگی:</p>			
ردیف	مرحله کار	حداقل نمره قبولی از ۳	نمره هنرجو
۱	به کارگیری دستور شرطی در برنامه	۲	
۲	به کارگیری دستور شرطی متداخل در برنامه	۱	
۳	توسعه برنامه	۱	
شایستگی‌های غیرفنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش:		۲	
استدلال - زبان فنی رعایت ارگونومی دقت در تشخیص ساختار شرطی مناسب هنگام تبدیل الگوریتم به برنامه در مسئله مطرح شده			
میانگین نمرات			*

* حداقل میانگین نمرات هنرجو برای قبولی و کسب شایستگی، ۲ است.

منابع

- ۱- دفتر تألیف کتاب‌های درسی فنی و حرفه‌ای و کاردانش (۱۳۹۴)، برنامه درسی درس تولید محتوای الکترونیک و برنامه‌سازی رشته شبکه و نرم‌افزار رایانه.
- ۲- حافظی نژاد، محمد، (۱۳۹۵)، چهارگام اصلی تولید محتوای الکترونیکی.
- ۳- کارکن، مهناز و دیگران، (۱۳۹۴)، کتاب معلم راهنمای تدریس برنامه‌سازی یک سی‌شارپ، تهران: شرکت چاپ و نشر کتاب‌های درسی ایران.
- ۴- کربلایی، مجید، (۱۳۹۳)، برنامه‌سازی ۱ کد ۳۵۸/۷۰، تهران: شرکت چاپ و نشر کتاب‌های درسی ایران.
- ۵- محمدی، محمدرضا و مینایی، غلامرضا. (۱۳۷۷). نرم‌افزارهای چند رسانه‌ای کد ۴۵۱/۲. تهران: شرکت چاپ و نشر کتاب‌های درسی ایران.
- ۶- یاقوتی، نیلوفر. (۱۳۹۵) پنج مزیت ارائه محتوای آموزشی آنلاین. ۱۳۹۵/۷/۴.

7- Asadi, A (2015), Windows 10 for Beginners, Imagine Publishing Ltd, Volume(22), Issue 4, pages 28-33, 38-45

8 - James Madison University-Official Site.

9- Serif, Student Project Resources.

10- Adobe Captivate – Official Site, Blogs.adobe.com/captivate9

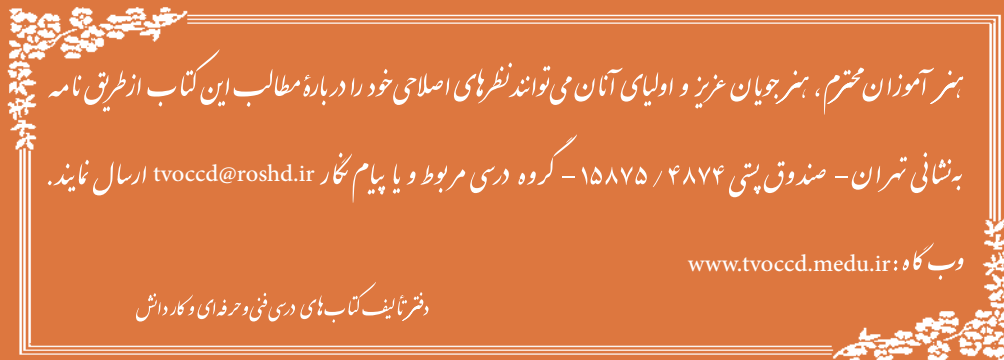
11- Multimedia Design: Navigation Maps and Storyboards. (2011)

12- Swish Max 4.0 user guide

13- Proshow Gold 6.0 help documentation

14- C# Programming Guide. [microsoft](http://microsoft.com)





هنرآموزان محترم، هنرجویان عزیز و اولیای آنان می‌توانند نظریه‌های اصلاحی خود را درباره‌ی مطالب این کتاب از طریق نامه برنشانی تهران - صندوق پستی ۴۸۷۴ / ۱۵۸۷۵ - گروه درسی مربوط و یا پیام‌نگار tvoccd@roshd.ir ارسال نمایند.

وب‌گاه: www.tvoccd.medu.ir

دفترتایپ کتاب‌های درسی فنی و حرفه‌ای و کارداش

همکاران هنرآموز که در فرایند اعتبارسنجی این کتاب مشارکت فعال داشته‌اند:
استان سمنان: اشراف‌السادات موسوی، سیمین معمارزاده
استان سیستان و بلوچستان: علیرضا حمیدی، هوشنگ ابراهیمی
استان فارس: بهاره بهره‌دار، سارا شیخی
استان کهگیلویه و بویراحمد: محمد فرجی‌زاده
استان گیلان: محمود باقری، حمیدرضا گل‌محمدی

