

درسنامه تولید محتوای الکترونیک و برنامه سازی
پودمان ۵ حل مسائل شرطی
واحد یادگیری ۹ شایستگی کار با عملگرهای ریاضی ، مقایسه ای و منطقی

تهیه و تنظیم: سکینه شاه محمدزاده

هدف از این درسنامه :

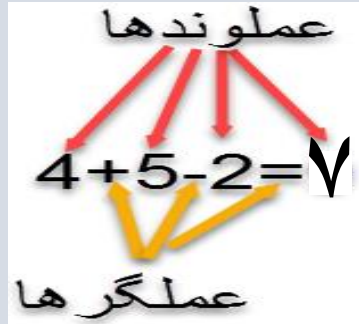
به کارگیری انواع عملگرها برای ایجاد عبارات محاسباتی در برنامه و محاسبه مقدار یک عبارت محاسباتی با توجه به اولویت عملگرها است.
جهت دانلود رایگان سوالات استخدامی هنرآموز شبکه و نرم افزار رایانه، اینجا بزنید.

یکی از کاربردهای اولیه رایانه، انجام عملیات ریاضی و مقایسه ای است. در زبان های برنامه نویسی برای انجام این عملیات از **عبارت** (expression) استفاده می شود.

یک عبارت حداقل از یک **عملگر** و یک یا چند **عملوند** تشکیل شده است.

مثال ۱: $x+y$ یک عبارت است که در آن x و y عملوند و علامت $+$ عملگر به حساب می آیند

مثال ۲: $result=10+12$ یک عبارت است که $result$ و 10 و 12 عملوند و $+$ و $=$ عملگر هستند.



عملگر (operator): به علامت هایی که بیانگر انجام یک عمل روی متغیرها و داده ها هستند، عملگر می گویند.

مثال عددی: $4+5-2=7$ در این مثال (جمع و تفریق و انتساب) عملگرها هستند.

مثال غیر عددی: $"Alireza" = "Ali" + "reza"$ در مثال فوق $+$ و $=$ عملگر هستند.



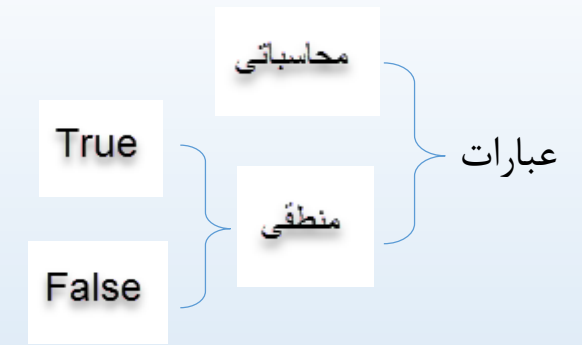
عملوند (operand): به مقادیر عددی و متغیرها، که یک عملگر روی آنها عملی را انجام می دهد، عملوند می گویند.

مثال عددی: $4+5-2=7$

در این مثال اعداد 4 و 5 و 2 و 7 عملوندهای عملگر جمع و تفریق و انتساب هستند.

مثال غیر عددی: $"Alireza" = "Ali" + "reza"$

در مثال فوق Ali و $reza$ عملوندهای عملگر $+$ هستند.



عبارات محاسباتی: عبارات محاسباتی شامل عملیات یک یا چند **عملگر** بر روی یک یا چند **عملوند** هستند. در نتیجه هر عبارت محاسباتی از تعدادی عملوند و عملگر تشکیل شده است که حاصل و نتیجه می تواند عددی یا غیر عددی باشد.

• مثال عددی: $4+5-2=7$

(در این مثال اعداد ۵ و ۴ و ۲ و ۷ عملوند هستند. و + و - و = عملگر هستند.)

• مثال غیر عددی: "Alireza" = "Ali" + "reza"

(در مثال فوق Ali و reza و Alireza عملوند و + و = عملگر هستند.)

بنابراین عملوندها اعدادی هستند که عملگرها بر روی آنها عمل می کنند .

عبارات منطقی یا بولین (Bool): به عباراتی که شامل عملگرهای مقایسه ای باشند، عبارات منطقی می گویند که بیان کننده یک شرط است .

جواب شرط یا نتیجه یک عبارت منطقی ، یکی از مقادیر True یا False می باشد

مثال: $a < b$

نمونه تمرین: در مثال زیر عملگرها و عملوندها را مشخص نمایید.

عملوند	عملگر	عبارت
۵	-	-۵
۷ ۱۳ ۱۲	- +	۷-۱۳+۱۲
a b ۲	- *	a-b*۲
۴۸ ۲ ۹	/ *	۴۸/۲*۹
A b c f	* / -	A*b/c-f
۲۵ ۵ ۲	/ *	۲۵/۵*۲

عملگرها در سی شارپ می توانند بر اساس تعداد عملوندهایی که روی آن ها عملیات انجام می شود به سه گروه دسته بندی شوند:

عملگرهای سه تایی	عملگرهای دوتایی	عملگرهای یکتایی
عملگرهایی که به سه عملوند نیاز دارند (خارج از بحث کتاب)	عملگرهایی که به دو عملوند نیاز دارند	عملگرهایی که به یک عملوند نیاز دارند
عملگر شرطی (: و ؟)	محاسباتی انتساب مقایسه ای منطقی	قرینه افزایشی کاهشی منطقی نقیض (not)

عملگرهای محاسباتی

عملگرهای محاسباتی: عملگرهای محاسباتی عملگرهایی هستند که اعمال محاسباتی را روی عملوندها انجام می دهند .

عملگر - : عملیات قرینه را انجام می دهد.

عملگر * : عملیات ضرب را انجام می دهد:

عملگر / : عملیات تقسیم را انجام می دهد: (این عملگر عملوند اول را بر عملوند دوم تقسیم می کند (تقسیم صحیح) و خارج قسمت صحیح را بر می گرداند)

عملگر % : باقیمانده تقسیم دو عدد را برای ما بر می گرداند(این عملگر عملوند اول را بر عملوند دوم تقسیم می کند و باقی مانده را بر می گرداند)

عملگر + : برای جمع دو عدد استفاده می شود.

عملگر - : عملیات تفریق را انجام می دهد.

مثال زیر برای مقادیر $b=7$ و $c=2$ محاسبه شده است.

عملگرها	نام	مثال	نتیجه
-	قرینه	$A=-b;$	$A=-7$
*	ضرب	$A=b*c;$	$A=14$
/	تقسیم	$A=b/c;$	$A=3$
%	باقیمانده تقسیم	$A=b\%c;$	$A=1$
+	جمع	$A=b+c;$	$A=9$
-	تفریق	$A=b-c;$	$A=5$

با توجه به آمار سال ۹۰ کشورمان ، برنامه ای بنویسید که جمعیت مردم ایران و تعداد خانوار را دریافت کرده ، بعد خانوار را در ایران نمایش دهد.
توضیح برنامه:

- دو متغیر به نام های `populate` و `numfamily` از نوع داده `long` برای جمعیت و تعداد خانوار تعریف کرده ایم.
 - متغیری به نام `familysize` از نوع اعشاری `float` برای بعد خانوار تعریف کرده ایم.
 - پیغامی مناسبی برای کاربر نمایش می دهد.(پیغام اختیاری است و می توان آن را ننوشت).
 - پیغام "Enter Population:" اطلاع می دهد که باید عددی را برای جمعیت وارد کنیم.
 - در این سطر با دستور `Readline()` هر آنچه که از کاربر دریافت میکند به عنوان رشته می شناسد و با دستور `parse` نوع داده ، رشته را به عدد تبدیل کرده و به عنوان عدد می شناسد.
 - این دستور اطلاع می دهد که باید عددی را برای تعداد خانوار وارد کنیم.
 - عدد را از کاربر دریافت کرده در متغیر `numfamily` قرار می دهد.
 - بعد خانوار برابر است با جمعیت تقسیم بر تعداد خانوار
 - با دستور `Console.WriteLine` جانگهدار بعد خانوار در خروجی نمایش داده می شود.
- ```
{
 long populate, numfamily;
 float familysize;
 Console.WriteLine("----- family size in IRAN -----");
 Console.WriteLine("Enter Population:");
 populate = long.Parse(Console.ReadLine());
 Console.WriteLine("Enter number of family :");
 numfamily = long.Parse(Console.ReadLine());
 familysize = populate / numfamily;
 Console.WriteLine("family size in IRAN: {0}", familysize);
 Console.ReadKey();
}
```

### خروجی برنامه فوق:

همانطور که مشاهده کردید در برنامه قبلی خروجی برای اعداد مختلف جمعیت و تعداد خانوار فقط عدد ۳ را نشان می دهد در حالی که عملا پاسخ چنین نیست و در محاسبه با ماشین حساب پاسخ دیگری بدست می آید. به علت اینکه هر دو طرف تقسیم را از نوع صحیح تعریف کردیم و برای رفع مشکل باید حداقل یکی از عملوندهای اطراف تقسیم را از نوع اعشاری تعریف کنیم.

| خروجی | جواب شما          | تعداد خانوار | جمعیت    |
|-------|-------------------|--------------|----------|
| ۳     | 3.547197260484893 | ۲۱۱۸۵۶۴۷     | ۷۵۱۴۹۶۶۹ |
| ۳     | 3.392691524854691 | ۹۵۳۱۴        | ۳۲۳۳۷۱   |
| ۳     | 3.793296991722189 | ۲۴۷۶۵        | ۹۳۹۴۱    |

```
file:///C:/Users/Hadiseh/Desktop/ConsoleApplication1/Consol
----- family size in IRAN -----
Enter Population:
75149669
Enter number of family :
21185647
family size in IRAN: 3
```

```
file:///C:/Users/Hadiseh/Desktop/ConsoleApplication1/Consol
----- family size in IRAN -----
Enter Population:
323371
Enter number of family :
95314
family size in IRAN: 3
```

```
file:///C:/Users/Hadiseh/Desktop/ConsoleApplication1/Cons
----- family size in IRAN -----
Enter Population:
93941
Enter number of family :
24765
family size in IRAN: 3
```



برای حل مشکل : برنامه قبل را به صورت زیر تغییر می دهیم تا جواب دقیق اعشاری بدست آید.

```
{
 long numfamily;
 float familysize, populate;
 Console.WriteLine("----- family size in IRAN -----");
 Console.WriteLine("Enter Population:");
 populate = long.Parse(Console.ReadLine());
 Console.WriteLine("Enter number of family :");
 numfamily = long.Parse(Console.ReadLine());
 familysize = populate / numfamily;
 Console.WriteLine("family size in IRAN: {0}", familysize);
 Console.ReadKey();
}
```

```
file:///C:/Users/Hadiseh/Desktop/ConsoleApplication1/Cons
----- family size in IRAN -----
Enter Population:
75149669
Enter number of family :
21185647
family size in IRAN: 3.547197
```

```
file:///C:/Users/Hadiseh/Desktop/ConsoleApplication1/ConsoleApp
----- family size in IRAN -----
Enter Population:
323371
Enter number of family :
95314
family size in IRAN: 3.392692
-
```

```
file:///C:/Users/Hadiseh/Desktop/ConsoleApplication1/Co
----- family size in IRAN -----
Enter Population:
93941
Enter number of family :
24765
family size in IRAN: 3.793297
```

**نکته کارگاه ۱:** نتیجه عملگر تقسیم با توجه به نوع عملوند هایش می تواند صحیح یا اعشاری باشد.

اگر حداقل یکی از عملوندهای عملگر تقسیم، از نوع داده اعشاری باشد، نتیجه عبارت، عدد اعشاری و در غیر اینصورت عدد صحیح است.

مثال:

```
{
 int a;
 a = 10 / 3;
 Console.WriteLine(a);
 Console.ReadLine();
}
```

خروجی

3

```
{
 double a;
 a = 10 / 3.2;
 Console.WriteLine(a);
 Console.ReadLine();
}
```

خروجی

3.125

در این مثال هر دو عدد ۱۰ و ۳ عدد صحیح هستند پس خروجی هم عدد صحیح خواهد بود.

در این مثال چون یکی از اعداد اعشاری است (۳/۲) پس خروجی هم عدد اعشاری خواهد بود.

## اولویت بندی عملگرهای محاسباتی (حق تقدم عملگرها):

منظور از اولویت عملگرها چیست؟ در عبارتی که چندین عملگر در آن استفاده شده باشد، انجام عملیات ها بر اساس حق تقدم عملگرها انجام می شود.  
اگر عبارت بیش از یک عملگر داشته باشد کدام عملگر را اول انجام بدهیم؟

می خواهیم حاصل عبارت زیر را بدست آوریم:

$$2+3*4=?$$

۱

$$2+3 *4=$$

۵

$$*4=20$$

روش اول: در این مثال ابتدا ۲ با ۳ جمع می شود (۵) و پاسخ در ۴ ضرب می شود که جواب ۲۰ بدست می آید

۱

$$2+ 3*4 =$$

۱۲

$$2+ 12 = 14$$

روش دوم : اول ۳ در ۴ ضرب و سپس پاسخ بدست آمده با ۲ جمع می شود که پاسخ ۱۴ بدست می آید.

بنابراین با توجه به جواب های مختلف در یک عبارت محاسباتی با تعداد عملگرهای بیشتر، باید حق تقدم عملگرها رعایت شود.

## جدول اولویت عملگرها

| درجه اولویت | عملگر         | عملکرد                          | نوع عملگر |
|-------------|---------------|---------------------------------|-----------|
| ۱           | -<br>++<br>-- | قرینه<br>افزایشی<br>کاهشی       | یکتایی    |
| ۲           | *<br>/<br>%   | ضرب<br>تقسیم<br>باقیمانده تقسیم | دوتایی    |
| ۳           | +<br>-        | جمع<br>تفریق                    | دوتایی    |

## نکات اولویت بندی عملگرها

مثال:

$$4 * 3 / 2 = ?$$

$$12 / 2 = 6$$

**نکته:** اگر دو یا چند عملگر دارای اولویت یکسان باشند اولویت سمت چپی بیشترین اولویت را دارد بنابراین طبق این نکته در مثال فوق چون عملگر ضرب و تقسیم در یک اولویت قرار دارند پس عملیات را به ترتیب از سمت چپ انجام می دهیم.

مثال:

$$4 * 3 / (2 + 1) = ?$$

$$4 * 3 / (2 + 1) = 4 * 3 / 3 = 12 / 3 = 4$$

**نکته:** محاسبات داخل پرانتز بیشترین اولویت را دارند.

مثال:

$$4 * (2 + (3 + 1)) = ?$$

$$4 * (2 + 4) =$$

$$4 * 6 = 24$$

**نکته:** در صورتی که چندین پرانتز داخل یکدیگر باشند بیشترین اولویت با داخلی ترین پرانتز خواهد بود.

مثال ۴:

$$14 \% 3 = 2$$

$$\begin{array}{r} 14 \phantom{0} \\ - 12 \phantom{0} \\ \hline 2 \phantom{0} \end{array}$$

**نکته:** منظور از علامت % باقیمانده تقسیم می باشد.

برای سه کالای اولیه، تخفیف ۱۰ درصد و تخفیف کالای چهارم ۷۰ درصد در نظر گرفته شده است. برنامه ای بنویسید که قیمت قبل از تخفیف ۴ کالا را از ورودی دریافت کرده، مبلغ تخفیف را محاسبه کند.

تعریف متغیر های k1,k2,k3,k4 از نوع int برای چهار کالا

چاپ پیغام Final Price برای کاربر

در خط جدید چاپ پیغام Enter price1: (وارد کنید قیمت کالای ۱ را)

دریافت قیمت کالا از کاربر در ورودی به وسیله متد console.ReadLine()

تعریف متغیر price1 از نوع float برای میزان تخفیف کالای اول که برابر است باقیمت کالای اول ضربدر تخفیف ۱۰ درصد تقسیم بر ۱۰۰

در خط جدید چاپ پیغام Enter price2: (وارد کنید قیمت کالای ۲ را)

دریافت قیمت کالا از کاربر در ورودی به وسیله متد console.ReadLine()

تعریف متغیر price2 از نوع float برای میزان تخفیف کالای دوم که برابر است باقیمت کالای دوم ضربدر تخفیف ۱۰ درصد تقسیم بر ۱۰۰

در خط جدید چاپ پیغام Enter price3: (وارد کنید قیمت کالای ۳ را)

دریافت قیمت کالا از کاربر در ورودی به وسیله متد console.ReadLine()

تعریف متغیر price3 از نوع float برای میزان تخفیف کالای سوم که برابر است باقیمت کالای سوم ضربدر تخفیف ۱۰ درصد تقسیم بر ۱۰۰

در خط جدید چاپ پیغام Enter price4: (وارد کنید قیمت کالای ۴ را)

دریافت قیمت کالا از کاربر در ورودی به وسیله متد console.ReadLine()

تعریف متغیر price4 از نوع float برای میزان تخفیف کالای چهارم که برابر است باقیمت کالای چهارم ضربدر تخفیف ۷۰ درصد تقسیم بر ۱۰۰

نمایش میزان تخفیف کالای ۱

نمایش میزان تخفیف کالای ۲

نمایش میزان تخفیف کالای ۳

نمایش میزان تخفیف کالای ۴

چاپ و نمایش مجموع تخفیف ها

```
{
int k1, k2, k3, k4;
Console.WriteLine("----- Final Price-----");
Console.Write("\nEnter Price1:");
k1 = int.Parse(Console.ReadLine());
float price1 = k1*10/100;
Console.Write("\nEnter Price2:");
k2 = int.Parse(Console.ReadLine());
float price2 = k2*10/100;
Console.Write("\nEnter Price3:");
k3 = int.Parse(Console.ReadLine());
float price3 =k3*10/100;
Console.Write("\nEnter Price4:");
k4 = int.Parse(Console.ReadLine());
float price4 =k4*70/100;
Console.WriteLine(price1);
Console.WriteLine(price2);
Console.WriteLine(price3);
Console.WriteLine(price4);
Console.WriteLine(price1 + price2 + price3 + price4);
Console.ReadLine();
}
```

خروجی برنامه فوق:

```
----- Final Price-----
Enter Price1:200
Enter Price2:300
Enter Price3:100
Enter Price4:500
20
30
10
350
410
```

ابتدا قیمت کالاها را به ترتیب از کار بر دریافت می کند.

سپس تخفیف کالاهای ۱ و ۲ و ۳ را با محاسبه ۱۰ درصد و کالای چهارم با محاسبه تخفیف ۷۰ درصد به کاربر نشان می دهد.

در پایان مجموع قیمت تخفیف ها را نمایش می دهد.

## نکات کارگاه : دنباله معنی دار

نکته: از دنباله معنی دار می توان در دستور `write` و `writeline` استفاده کرد برای اینکه یک خروجی بهتری داشته باشیم.

مثال: `\n` یک خط خالی در خروجی ایجاد می کند. که می توان این خط خالی را به همراه پیغام مناسب برای کاربر ایجاد کرد.

```
{
 Console.Write("hello");
 Console.WriteLine(2 + 4);
 Console.ReadLine();
}
```

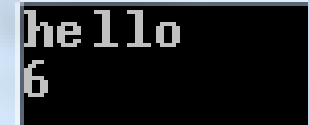
خروجی



```
hello6
```

```
{
 Console.Write("hello \n");
 Console.WriteLine(2 + 4);
 Console.ReadLine();
}
```

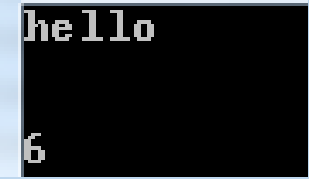
خروجی



```
hello
6
```

```
{
 Console.Write("hello \n \n \n");
 Console.WriteLine(2 + 4);
 Console.ReadLine();
}
```

خروجی



```
hello

6
```



## نکات کارگاه : دنباله معنی دار

مثال : \t کار tab را انجام می دهد و بین کلمات فاصله ایجاد می کند.

```
{
 Console.WriteLine("honarestan mahdieh in Ahar");
 Console.WriteLine("honarestan \t mahdieh \t in \t Ahar");
 Console.ReadLine();
}
```

خروجی

```
honarestan mahdieh in Ahar
honarestan mahdieh in Ahar
```

جدول عملکرد برخی دنباله های فرار:

| عملکرد دنباله                        | دنباله معنی دار |
|--------------------------------------|-----------------|
| ایجاد یک بوق هشدار(Alarm)            | \a              |
| حذف یک کاراکتر(Backspace)            | \b              |
| بازگشت به ابتدای خط(carriage Return) | \r              |
| ایجاد یک خط خالی(new line)           | \n              |
| ایجاد یک tab یا فاصله افقی(Tab)      | \t              |
| ایجاد یک تک کوتیشن                   | \'              |
| ایجاد یک دابل کوتیشن                 | \"              |
| ایجاد یک بک اسلش(Backslash)          | \\              |

با عملگرهای ساده همچون +، -، \*، / و..... در محیط سی شارپ آشنا شدید.

اما عملگرهایی مانند  $\sqrt{\quad}$ ،  $\wedge$  و... که علامتی برای استفاده مستقیم در C# ندارند بلکه یکسری متدهایی برای آنها در C# تعریف شده و داخل کلاس Math قرار گرفته است و با استفاده از این متدها می توان محاسبات پیشرفته ریاضی را انجام داد.

### برخی از متدهای کلاس Math

| نام متد | شرح متد                                   | نوع داده ورودی | نوع داده خروجی | معادل ریاضی                        | معادل C#            | پاسخ عبارت |
|---------|-------------------------------------------|----------------|----------------|------------------------------------|---------------------|------------|
| Pow     | مقدار عدد به توان را برمی گرداند          | double         | double         | $2^5$                              | Math.Pow(2,5)       | 32         |
| Sqrt    | ریشه دوم (جذر) عدد را برمی گرداند         | double         | double         | $\sqrt{9}$                         | Math.Sqrt(9)        | 3          |
| Round   | عدد را با توجه به تعداد اعشار گرد می کند. | decimal        | decimal        | گرد کردن عدد ۵/۶۱۹ تا دو رقم اعشار | Math.Round(5.619,2) | 5.62       |

**نکته:** در متد Math.Sqrt() ورودی نمی تواند عدد منفی باشد.

**نکته:** در متد Math.Round(5.619.2) : برای گرد کردن عدد ۵/۶۱۹ آخرین رقم بعد از انتخاب تعداد رقم اعشار یعنی بعد از دو رقم اعشار ۹ اگر بزرگتر از ۵ بود عدد گرد شده یک واحد اضافه می شود و عدد ۶۲ بدست می آیدولی اگر کمتر از ۵ باشد عدد گرد شده تغییری نمی کند و عدد ۶۱ بدست می آید .

مثال : گرد شده عدد ۵/۶۱۹ تا دو رقم اعشار برابر است با ۵/۶۲

گرد شده عدد ۵/۶۱۲ تا دو رقم اعشار برابر است با ۵/۶۱

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.

نکته: محیط دایره برابر است با  $2\pi r$   
محیط دایره:  $2 * \text{Math.PI} * r$  ( شعاع دایره)

توضیح برنامه:

```
Console.Write("Enter the radius in meter: ");
int radius = int.Parse(Console.ReadLine());
double env = 2 * Math.PI * radius;
Console.WriteLine("\n\n circle environment={0}", env);
```

- با دستور write به کاربر اعلام می کند که مقدار شعاع را بر حسب متر وارد کند.
- با دستور Readline() مقدار شعاع را از کاربر می گیرد.
- محاسبه محیط دایره ( $2\pi r$ ) و حاصل را داخل env قرار می دهد.
- env متغیری برای محیط دایره تعریف کرده ایم از نوع double
- محیط محاسبه شده را به وسیله جانگهدار به کاربر نمایش می دهد.

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

**نکته:** مساحت مربع برابر است با یک ضلع ضربدر خودش یا اندازه ضلع مربع به توان ۲ ← اندازه ضلع مربع = جذر مساحت

توضیح برنامه:

- با دستور `console.write` به کاربر اعلام می کند که مساحت مربع را وارد کند.
- متغیر `square_area` را از نوع `double` تعریف کرده ایم که مساحت مربع را با دستور `console.ReadLine()` دریافت کرده و داخل آن قرار می دهد.
- متغیر `Side` را از نوع `double` تعریف کرده ایم که جذر مساحت بدست آمده را به عنوان اندازه ضلع مربع داخل `Side` قرار می دهد.
- مقدار `Side` را به وسیله جانگهدار به کاربر نمایش می دهد.

```
/**3.....*/
Console.Write("\n\n enter the square area: ");
double square_area = double.Parse(Console.ReadLine());
double side = Math.Sqrt(square_area);
Console.Write("\n square side size={0}", side);
```

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.

**نکته :** مساحت مستطیل برابر است با طول \* عرض

توضیح برنامه:

- متغیری به نام s\_area از نوع double برای مساحت تعریف کرده ایم که برابر است با طول \* عرض
- متغیری به نام count از نوع double برای تعداد سرامیک ها تعریف کرده ایم که برابر است با مساحت کل مربع تقسیم بر مساحت سرامیک ها
- با دستور console.write تعداد سرامیک ها را نمایش می دهد.

```
/**4.....*/
double s_area = 30 * 60;
double count = square_area / s_area;
Console.WriteLine("\n \n number of tiles 30*60:{0}", count);
```

به 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.

**نکته :** مساحت مربع برابر است با یک ضلع به توان ۲

توضیح برنامه:

- در قسمت اول این برنامه مقدار شعاع را بر حسب متر محاسبه کردیم ولی در اینجا اندازه ها بر حسب سانتی متر است. بنابراین این با ضرب کردن محیط در ۱۰۰ مقدار محیط بدست آمده را تبدیل به سانتی متر می کنیم.
- تعریف متغیر S1\_area از نوع double برای مساحت کاشی
- تعریف متغیر area از نوع double برای مساحت مستطیل که برابر است با طول ضربدر عرض
- تعریف متغیری بنام numbertile از نوع double برای تعداد کاشی ها که برابر است با مساحت مستطیل تقسیم بر مساحت کاشی
- با دستور console.write با پیام مناسبی تعداد کاشی ها به کاربر نمایش داده می شود.

```
/**5.....**/
env = env * 100;
double s1_area = 15 * 15;
double area = env * 30;
double numbertile = area / s1_area;
Console.WriteLine("\n\n number of tiles 15*15:{0}", numbertile);
```

```
{
 Console.WriteLine("Enter the radius in meter: ");
 int radius = int.Parse(Console.ReadLine());
 double env = 2 * Math.PI * radius;
 Console.WriteLine("\n\n circle environment={0}", env);
 /*3.....*/
 Console.WriteLine("\n\n enter the square area: ");
 double square_area = double.Parse(Console.ReadLine());
 double side = Math.Sqrt(square_area);
 Console.WriteLine("\n square side size={0}", side);
 /*4.....*/
 double s_area = 30 * 60;
 double count = square_area / s_area;
 Console.WriteLine("\n \n number of tiles 30*60:{0}", count);
 /*5.....*/
 env = env * 100;
 double s1_area = 15 * 15;
 double area = env * 30;
 double numbertile = area / s1_area;
 Console.WriteLine("\n\n number of tiles 15*15:{0}", numbertile);
 Console.ReadKey();
}
```

```
Enter the radius in meter: 20

circle environment=125.663706143592

enter the square area: 240000

square side size=489.897948556636

number of tiles 30*60:133.333333333333

number of tiles 15*15:1675.51608191456
```

کل برنامه کارگاه ۳ یکجا

خروجی برنامه :


- شعاع را بر حسب متر وارد کنید.
- محیط دایره برابر است با
- مساحت مربع نگارخانه را وارد کنید بر اساس سانتی متر مربع
- اندازه ضلع مربع =
- تعداد کاشی  $30 * 60 =$
- تعداد کاشی  $15 * 15 =$



## عملگرهای انتساب

به مثال های زیر توجه کنید:


عملگر انتساب =  $X = Y$  ;



• در عبارت فوق مقدار متغیر سمت راست  $Y$  در متغیر سمت چپ  $X$  قرار می گیرد. (مثال:  $Y=2$  آنگاه  $X=2$ )

• در عبارت  $X=Y=Z=0$  سه تا عملگر انتساب وجود دارد که از سمت راست محاسبه می شوند (مقدار ۱۰ در متغیر  $Z$  قرار می گیرد و سپس مقدار  $Z$  همان ۱۰ در متغیر  $Y$

قرار گرفته و در آخر مقدار متغیر  $Y$  در  $X$  قرار می گیرد. پس مقدار  $X$  برابر با ۱۰ می باشد.)



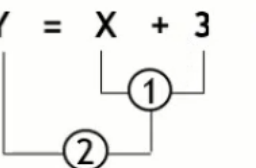
$$X = Y = Z ;$$

(2)      (1)

$$X + Y + Z ;$$

(1)      (2)

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

$$Y = X + 3 ;$$


**نکته:** اگر در عبارتی هم عملگر محاسباتی داشته باشیم و هم عملگر انتساب، اولویت عملگرهای محاسباتی از عملگرهای انتساب بیشتر است.

## عملگرهای انتساب

- از ترکیب عملگرهای محاسباتی و علامت = مجموعه دیگری از عملگرها ایجاد می شود که عمل محاسباتی و انتساب را انجام دهند .
- به عبارت دیگر این نوع از عملگرها مقدار و نتیجه عبارت سمت راست خود را در متغیر سمت چپ قرار می دهند.
- **عملگر =** عملیات انتساب ساده را انجام می دهد. یعنی مقدار سمت راست را داخل عملوند سمت چپی قرار می دهد.
  - **عملگر +=** این عملگر، مقدار سمت راست را به مقدار موجود عملوند سمت چپ اضافه کرده و نتیجه را داخل عملوند سمت چپ می ریزد:
  - **عملگر -=** این عملگر، مقدار سمت راست را از مقدار موجود عملوند سمت چپ کم کرده و نتیجه را داخل عملوند سمت چپ میریزد:
  - **عملگر \*=** این عملگر، مقدار سمت راست را در مقدار موجود عملوند سمت چپ ضرب کرده و نتیجه را داخل عملوند سمت چپ میریزد
  - **عملگر /=** این عملگر، مقدار سمت راست را از مقدار موجود عملوند سمت چپ تقسیم کرده و نتیجه را داخل عملوند سمت چپ میریزد:
  - **عملگر %=** این عملگر، باقیمانده تقسیم عملوند سمت چپ بر مقدار سمت راست را در عملوند سمت چپ میریزد:

| مثال     | نام              | عملگرها |
|----------|------------------|---------|
| $A=b;$   | مساوی            | =       |
| $A+=b;$  | انتساب جمع       | +=      |
| $A-=b;$  | انتساب تفریق     | -=      |
| $A*=b;$  | انتساب ضرب       | *=      |
| $A/=b;$  | انتساب تقسیم     | /=      |
| $A\%=b;$ | انتساب باقیمانده | \%=     |

## عملگرهای انتساب

مثال:

حل جدول صفحه ۲۲۶ کتاب با توجه به مقادیر `int X=10, Y=3`

| عبارت              | معادل               | مقدار دهی           | نتیجه X |
|--------------------|---------------------|---------------------|---------|
| <code>X+=y;</code> | <code>X=x+y;</code> | <code>X=10+3</code> | ۱۳      |
| <code>X-=y;</code> | <code>X=x-y;</code> | <code>X=10-3</code> | ۷       |
| <code>X*=y;</code> | <code>X=x*y;</code> | <code>X=10*3</code> | ۳۰      |
| <code>x/=y;</code> | <code>X=x/y;</code> | <code>X=10/3</code> | ۳       |
| <code>X%=y;</code> | <code>X=x%y;</code> | <code>X=10%3</code> | ۱       |

## عملگرهای افزایشی - کاهششی

**عملگر ++:** این عملگر به مقدار موجود یک عدد اضافه می کند (افزایش متغیر به میزان یک واحد)

**عملگر --:** این عملگر از مقدار موجود یک عدد کم می کند. (کاهش متغیر به میزان یک واحد)

در این مثال برای مقدار  $b = 4$  نتیجه زیر حاصل می شود.

| نتیجه | مثال      | نام     | عملگرها |
|-------|-----------|---------|---------|
| $A=4$ | $A=b++;$  | افزایشی | ++      |
| $A=5$ | $A=++b;$  | افزایشی | ++      |
| $A=4$ | $A=b--;$  | کاهششی  | --      |
| $A=3$ | $A=---b;$ | کاهششی  | --      |

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

توضیح برنامه:

```
{
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);
Console.ReadKey();
}
```

- تعریف متغیر count101 برای تعداد دانش آموزان کلاس ۱۰۱ و متغیر count 102 برای تعداد دانش آموزان کلاس ۱۰۲
- با پیام مناسبی عددی را به عنوان تعداد دانش آموزان کلاس ۱۰۱ از کاربر دریافت کرده و در متغیر count101 قرار می دهد
- با پیام مناسبی عددی را به عنوان تعداد دانش آموزان کلاس ۱۰۲ از کاربر دریافت کرده و در متغیر count102 قرار می دهد
- با استفاده از عبارت جایگزینی مقابل عبارت count of students class 101 مقدار count101-- را قرار می دهد.
- با استفاده از دستور جایگزینی حاصل count 101 را که یک واحد کاهش داده نمایش می دهد.
- با استفاده از عبارت جایگزینی مقابل عبارت count of students class 102 مقدار count102++ را قرار می دهد.
- با استفاده از دستور جایگزینی حاصل count 102 را که یک واحد کاهش داده نمایش می دهد.

خروجی

```
Enter count of student class 101:14
Enter count of student class 102:23
count of students class 101=14
count101=13
count of students class 102=23
count102=24
```

فرق برنامه زیر با برنامه صفحه قبلی :

```
{
 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);
 Console.ReadKey();
}
```

خروجی

```
Enter count of student class 101:14
Enter count of student class 102:23
count of students class 101=13
count101=13
count of students class 102=24
count102=24
```

## عملگر الحاق دورشته

مثال: به برنامه مقابل دقت کنید.

توضیح برنامه:

```
{
int a = 5 + 7;
string s = "ali" + "reza";
string count = "count of pencil:" + 100;
Console.WriteLine(a);
Console.WriteLine(s);
Console.WriteLine(count);
Console.ReadKey();
}
```

• تعریف متغیر a از نوع int که برابر است با ۵+۷

• تعریف متغیر S از نوع string با مقدار "ali" + "reza"

(مقدار S برابر است با مجموع دو مقدار رشته ای)

• تعریف متغیر count از نوع string که برابر است با

مجموع دو مقدار رشته ای (count of pencil:) و (۱۰۰)

• چاپ مقدار a

• چاپ مقدار S

• چاپ مقدار count

خروجی:

• چاپ مقدار a که برابر است با مجموع دو عدد ۵+۷

• چاپ مقدار S که برابر است با الحاق دو رشته "ali" + "reza"

• چاپ مقدار count که برابر است با الحاق دو رشته "count of pencil" و ۱۰۰

( ۱۰۰ ) را به عنوان رشته می شناسد چون از نوع string تعریف کرده ایم.)

```
12
alireza
count of pencil:100
```

## عملگر الحاق دورشته

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

• تعریف متغیر S از نوع string با مقدار رشته "ali"

• از عملگر += استفاده کرده ایم تا مقدار قبلی S را با مقدار "reza" جمع کند و الحاق

( S+="reza" معادل است با ( s=s+"reza"

• چاپ مقدار S

خروجی: الحاق دو رشته

alireza

**نکته:** اگر حداقل یکی از عملوند های عملگر + از نوع string باشد ، این عملگر به عنوان عملگر الحاق دورشته عمل می کند.

**نکته:** عملگر + برای الحاق دو رشته و عملگر += به عنوان عملگر انتساب الحاق دو رشته به کار می رود.



**نکته :** عملگرهای مقایسه ای برای مقایسه بین دو عملوند مورد استفاده قرار می گیرند. نتیجه این مقایسه یک عبارت بولی (boolean) یا True و False می باشد. باید دقت کنید که اگر نتیجه مقایسه ۲ مقدار درست باشد True و در غیر این صورت False را برمیگرداند.

از این دسته از عملگرها در دستورات شرطی بیشتر استفاده می شود.

- **عملگر ==:** این عملگر بررسی می کند که دو عبارت با هم برابر هستند یا خیر. در صورت برابر بودن مقدار true و در غیر اینصورت مقدار false بر می گرداند:
  - **عملگر !=:** این عملگر بررسی می کند که دو عبارت مخالف یکدیگر هستند یا خیر:
  - **عملگر <:** این عملگر در صورتی true بر میگرداند که عملوند سمت چپ کوچکتر از عملوند سمت راست باشد در غیر اینصورت مقدار false بر می گرداند:
  - **عملگر <=:** این عملگر در صورتی true بر میگرداند که عملوند سمت چپ کوچکتر یا مساوی عملوند سمت راست باشد در غیر اینصورت مقدار false بر می گرداند:
  - **عملگر >:** این عملگر در صورتی true بر میگرداند که عملوند سمت چپ بزرگتر از عملوند سمت چپ باشد:
  - **عملگر >=:** این عملگر در صورتی true بر میگرداند که عملوند سمت چپ بزرگتر یا مساوی از عملوند سمت راست باشد:
- در مثال زیر برای مقادیر  $b=4$  و  $c=2$  و A از نوع bool حاصل بدست آمده است.

| نتیجه   | مثال    | نام              | معادل ریاضی | عملگرها |
|---------|---------|------------------|-------------|---------|
| A=false | A=b==c; | متساوی یا برابری | =           | ==      |
| A=true  | A=b!=c; | نا مساوی         | ≠           | !=      |
| A=false | A=b<c;  | کوچکتر           | <           | <       |
| A=true  | A=b>c   | بزرگتر           | >           | >       |
| A=false | A=b<=c; | کوچک تر مساوی    | ≤           | <=      |
| A=true  | A=b>=c; | بزرگ تر مساوی    | ≥           | >=      |

## عملگرهای مقایسه ای

مقایسه دو عبارت  $A=B$  و  $A==B$

**نکته:** در عبارت  $A=B$  علامت  $=$  به عنوان عملگر انتساب عمل می کند و مقدار متغیر  $B$  را در متغیر  $A$  قرار می دهد.

**نکته:** در عبارت  $A==B$  علامت  $==$  به عنوان عملگر مقایسه ای است و عبارت یک عبارت منطقی و نتیجه  $True$  یا  $False$  خواهد بود

**نکته:** اولویت عملگرهای محاسباتی بالاتر از عملگرهای مقایسه ای است.

برنامه ای بنویسید که سرانه مصرف آب در ایران را با سرانه مصرف آب در دنیا مقایسه کند.

توضیح برنامه:

```
{
int iranWater = 190;
Console.WriteLine(" The average consumption in IRAN -> " + iranWater);
int worldWater = 135;
Console.WriteLine(" The average consumption in WORLD -> " + worldWater);
//مقایسه مصرف آب ایران و جهان
Console.WriteLine(" Is iranWater > worldWater:{0}", iranWater > worldWater);
Console.ReadLine();
}
```

- تعریف متغیر iranWater از نوع int با مقداردهی ۱۹۰
- با دستور مناسب مقدار متغیر iranWater نمایش داده می شود.
- تعریف متغیر worldWater از نوع int با مقداردهی ۱۳۵
- مقدار worldWater با عبارت مناسب و نوشته شده نمایش داده می شود.
- با یک پیغام مناسب مقادیر دو متغیر iranWater و worldWater را باهم

مقایسه کرده اگر مقدار `iranWater > worldWater` خروجی True در غیر اینصورت False را برمی گرداند.

خروجی

```
The average consumption in IRAN -> 190
The average consumption in WORLD -> 135
Is iranWater > worldWater:True
```

خروجی:

- متوسط مصرف ایران ۱۹۰
- متوسط مصرف جهان ۱۳۵
- چون  $190 > 135$  پس خروجی True

برنامه ای بنویسید که عدم قبولی یک هنرجو را با نمایش **False** به علت گرفتن نمره ۷ نشان دهد.

توضیح برنامه:

```
{
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);
Console.ReadLine();
}
```

- تعریف متغیر Flag از نوع Bool و مقداردهی اولیه False
- تعریف متغیر Score از نوع int با مقدار اولیه ۷
- چاپ مقدار Score
- با عبارت مناسبی مقدار flag==true و score==7 بعد از بررسی با استفاده از

دستور جای گذاری چاپ می شود.

خروجی

```
score = 7
is student passed:False, is score=7:True
```

**نکته:** اولویت عملگرهای محاسباتی بالاتر از عملگرهای مقایسه ای است.

## انواع خطاها

### ۱- خطای زمان ترجمه یا کامپایل ( Compile error )

این خطاها زمانی تشخیص داده می شوند که دکمه F6 را بزنیم تا در ویژال استودیو برنامه را ترجمه کنیم. چند نمونه خطای زمان کامپایل:

- استفاده از کلمات کلیدی برای نام متغییر
- رعایت نکردن حساسیت به حروف کوچک و بزرگ
- استفاده از کنترلی که در فرم وجود ندارد.

### ۲- خطای زمان اجرا ( Runtime error )

خطاهایی که در زمان اجرای برنامه رخ می دهند. مانند تقسیم یک متغییر بر صفر

### ۱- خطای منطقی ( Logical error )

در این نوع خطاها کلا منطق برنامه مشکل دارد و به نتیجه دلخواه نمی رسیم.

می خواهیم عددی را دریافت کرده و 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

توضیح برنامه:

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

- تعریف متغیر num1 از نوع int
- پیام زیر روی صفحه نشان داده می شود.
- مقداری را بوسیله دستور console.ReadLine() و متد parse.نوع داده از کاربر دریافت کرده ، در متغیر num1 قرار می دهد.
- یک واحد به مقدار num1 می افزاید.
- مقدار num1 را در ۵ ضرب می کند.
- یک واحد از مقدار num1 کم می کند.
- مقدار num1 را بر ۲ تقسیم کرده و حاصل را به دست می آورد.
- مقدار متغیر num1 را با الگوی جایگذاری نمایش می دهد.

حال برنامه را با استفاده از دکمه F6 اجرا می کنیم که با این عمل لیست خطا در پایین صفحه به این صورت ظاهر می شود.

|   | Code   | Description                                                                                        | Project                 | File       | Line | Suppression St. |
|---|--------|----------------------------------------------------------------------------------------------------|-------------------------|------------|------|-----------------|
| ✘ | CS0201 | Only assignment, call, increment, decrement, and new object expressions can be used as a statement | ConsoleApplicatio<br>n3 | Program.cs | 17   | Active          |
| ✘ | CS0201 | Only assignment, call, increment, decrement, and new object expressions can be used as a statement | ConsoleApplicatio<br>n3 | Program.cs | 19   | Active          |

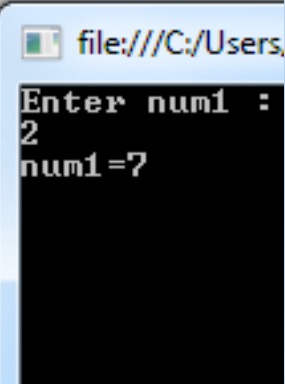
همانطور که مشاهده می کنید این برنامه دارای دو خطا می باشد. روی هر Error که دابل کلیک کنیم خطای مربوط به آن خط در برنامه به صورت های لایت نمایش داده می شود.

• اولین خطا و های لایت بعد از دابل کلیک روی سطر اول از خطاها در خط ۱۷ رخ داده است. و خطای  $num1 * 5$  است که این خطا اعلام می کند در برنامه نمی توان

$num1 * 5$  را به این صورت نوشت. و برای رفع خطا باید مقدار  $num1 * 5$  را داخل یک متغیر بریزیم و بنویسیم:  $Num1 = num1 * 5$

• و خطای دوم در خط ۱۹ به این علت ایجاد شده که  $num1 / 2$  را انجام دادیم ولی حاصل را به هیچ متغیری انتساب ندادیم. که برای رفع خطا می نویسیم:  $num1 = num1 / 2$

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



حال بعد از رفع خطاها اگر دکمه F6 را بزنیم دیگر خطایی نداریم و خروجی به صورت مقابل نمایش داده می شود.

برنامه مقابل را در سی شارپ نوشته و خطاهای آن را بررسی کنید.

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

| Error List      |        |                                                                           |                         |            |                      |                   |  |
|-----------------|--------|---------------------------------------------------------------------------|-------------------------|------------|----------------------|-------------------|--|
| Entire Solution |        | 4 Errors                                                                  | 0 Warnings              | 0 Messages | Build + IntelliSense | Search Error List |  |
|                 | Code   | Description                                                               | Project                 | File       | Line                 | Suppre            |  |
| ✘               | CS0019 | Operator '>' cannot be applied to operands of type 'bool' and 'int'       | ConsoleApplicatio<br>n5 | Program.cs | 14                   | Active            |  |
| ✘               | CS0019 | Operator '-' cannot be applied to operands of type 'string' and 'string'  | ConsoleApplicatio<br>n5 | Program.cs | 18                   | Active            |  |
| ✘               | CS0019 | Operator '-' cannot be applied to operands of type 'int' and 'string'     | ConsoleApplicatio<br>n5 | Program.cs | 19                   | Active            |  |
| ✘               | CS0019 | Operator '-=' cannot be applied to operands of type 'string' and 'string' | ConsoleApplicatio<br>n5 | Program.cs | 20                   | Active            |  |

بررسی خطاها:

- ۱- خطای خط ۱۴: نمی توان عملگر > را برای مقدار bool استفاده کرد.
- ۲- خطای خط ۱۸: نمی توان عملگر - را برای متغیر های رشته ای استفاده کرد.
- ۳- خطای خط ۱۹: نمی توان عملگر - را برای متغیر رشته ای (S1) استفاده کرد.
- ۴- خطای خط ۲۰: نمی توان عملگر -= را برای متغیر رشته ای استفاده کرد.



برنامه ای بنویسید که تعداد دفترچه ها و تعداد بچه های یک موسسه را از ورودی دریافت کرده ، تعیین کند به هر یک از بچه ها چند دفترچه تعلق می گیرد.

توضیح برنامه:

```
{
 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();
}
```

- تعریف متغیر booklet ( برای دفترچه ) و children ( بچه ها ) از نوع int
- با پیغام مناسبی تعداد دفترچه ها از ورودی دریافت می شود.
- با پیغام مناسبی تعداد بچه ها از ورودی دریافت می شود.
- تعداد دفترچه ها تقسیم بر تعداد بچه ها و قرار دادن مقدار آن داخل متغیر count
- نمایش عدد به دست آمده

دکمه F6 را می زنیم تا برنامه ترجمه شود. در ترجمه برنامه خطایی وجود ندارد.

```
file:///C:/Users/Hadiseh/Desktop/Cor
Enter number of booklet :
500
Enter number of children :
200
count=2
```

تعداد دفترچه ها: ۵۰۰  
تعداد بچه ها: ۲۰۰  
تعداد ۲ (یعنی به هر کدام از بچه ها ۲ تا دفترچه می رسد).

```
Enter number of booklet :
200
Enter number of children :
0
```

اجرای مجدد برنامه:  
تعداد دفترچه ها: ۵۰۰  
تعداد بچه ها: ۰

```
int count = booklet / children;
Console.WriteLine("count={0} ", c
Console.ReadKey();
```

**DivideByZeroException was unhandled**  
An unhandled exception of type 'System.DivideByZeroExcep  
ConsoleApplication5.exe  
Additional information: Attempted to divide by zero.

برنامه خطای مقابل را می دهد. و این خطا، خطای زمان اجراست.  
علت خطا: divide zero یعنی خطای تقسیم عدد بر صفر

عملگرهای منطقی بر روی عبارات منطقی عمل می کنند. عبارات منطقی دارای دو ارزش درستی یا نادرستی هستند. که در `#C` ارزش نادرستی با مقدار صفر و ارزش درستی با مقدار غیر صفر مشخص می شود.

**نتیجه عملگر !** این عملگر یا `NOT`، نتیجه یک عبارت `Boolean` را برعکس می کند، یعنی اگر یک عبارت نتیجه `true` برگرداند، این عملگر نتیجه را به `false` تبدیل می کند (عملگر نقیض ارزش عملوند را معکوس می کند)

**نتیجه عملگر &&** این عملگر یا `AND` در صورتی مقدار `True` بر می گرداند که هر دو عملوند مقدار `true` باشند. (وقتی درست است که هر دو عملوند ارزش درستی داشته باشد)

**نتیجه عملگر ||** این عملگر یا `OR` در صورتی مقدار `True` بر می گرداند که یکی از عملوندها نتیجه `true` داشته باشد (وقتی نادرست است که هر دو عملوند ارزش نادرستی داشته باشد (در بقیه موارد نتیجه آن ارزش درستی دارد.))

## جدول درستی یا نادرستی عملگرهای منطقی

| نتیجه         | مثال     | نام        | عملگرها |
|---------------|----------|------------|---------|
| A=false       | A=!b;    | نقیض (not) | !       |
| A=false       | A=b&& c; | و(And)     | &&      |
| <b>A=true</b> | A=b  c;  | یا(Or)     |         |

در این مثال برای مقادیر b=true و c=false برای مقدار A نتایج زیر حاصل می شوند.

| نتیجه          | مثال     | نام        | عملگرها |
|----------------|----------|------------|---------|
| A=true         | A=!b;    | نقیض (not) | !       |
| <b>A=false</b> | A=b&& c; | و(And)     | &&      |
| A=true         | A=b  c;  | یا(Or)     |         |

در این مثال برای مقادیر b=false و c=true برای مقدار A نتایج زیر حاصل می شوند.

| نتیجه          | مثال     | نام        | عملگرها |
|----------------|----------|------------|---------|
| <b>A=false</b> | A=!b;    | نقیض (not) | !       |
| A=true         | A=b&& c; | و(And)     | &&      |
| A=true         | A=b  c;  | یا(Or)     |         |

در این مثال برای مقادیر b=true و c=true برای مقدار A نتایج زیر حاصل می شوند.

| نتیجه         | مثال     | نام        | عملگرها |
|---------------|----------|------------|---------|
| <b>A=true</b> | A=!b;    | نقیض (not) | !       |
| A=false       | A=b&& c; | و(And)     | &&      |
| A=false       | A=b  c;  | یا(Or)     |         |

در این مثال برای مقادیر b=false و c=false برای مقدار A نتایج زیر حاصل می شوند.

رمز قفل یک چمدان مسافرتی ۲۵ است. می خواهیم برنامه ای بنویسیم که باز شدن قفل را شبیه سازی کند.  
( ارقام رمز را جداگانه از ورودی دریافت می کنیم.)

توضیح برنامه:

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

- تعریف دو متغیر به نام های X و Y یکی برای دریافت رقم اول و دیگری برای دریافت رقم دوم از نوع byte
- دریافت عدد از ورودی و قرار دادن آن داخل متغیر Y
- دریافت عدد از ورودی و قرار دادن آن عدد داخل متغیر X
- اگر مقدار X برابر ۲ و مقدار Y برابر ۵ باشد مقدار true را برمی گرداند و قفل باز می شود.  
در غیر اینصورت مقدار False را برمی گرداند و قفل باز نمی شود.

(و این همان شرط && (And) است و فقط زمانی درست است که هر دو عبارت درست باشد در غیر اینصورت نادرست است.)

```
5
2
True
```

```
5
3
False
```

```
5
5
False
```

```
4
2
False
```

خروجی برای اعداد مختلف:

قفل چمدان مسافرتی کارگاه ۱۰ خراب شده است و در صورت درست بودن حداقل یکی از ارقام رمز هم باز می شود. می خواهیم برنامه ای بنویسیم که باز شدن قفل را در این وضعیت شبیه سازی کند.

توضیح برنامه:

```
1
byte x, y;
y = byte.Parse(Console.ReadLine());
x = byte.Parse(Console.ReadLine());
Console.WriteLine((x == 2) || (y == 5));
Console.ReadKey();
}
```

• تعریف دو متغیر به نام های X و Y یکی برای دریافت رقم اول و دیگری برای دریافت رقم دوم از نوع byte

byte x, y;

y = byte.Parse(Console.ReadLine());

x = byte.Parse(Console.ReadLine());

Console.WriteLine((x == 2) || (y == 5));

Console.ReadKey();

• دریافت عدد از ورودی و قرار دادن آن داخل متغیر Y

• دریافت عدد از ورودی و قرار دادن آن عدد داخل متغیر X

• اگر مقدار X برابر ۲ یا مقدار Y برابر ۵ باشد مقدار true را برمی گرداند و قفل باز می شود

در غیر اینصورت مقدار False را برمی گرداند و قفل باز نمی شود.

(و این همان شرط || (یا or) است و فقط زمانی نادرست است که هر دو عبارت نادرست باشد در غیر اینصورت درست است.)

```
6
2
True
```

```
4
2
True
-
```

```
2
3
False
```

خروجی برای اعداد مختلف:

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

توضیح برنامه:

```
{
string weather;
Console.WriteLine("please enter weather:");
weather = Console.ReadLine();
Console.WriteLine("weather is not cloudy: {0}", weather != "cloudy");
Console.ReadKey();
}
```

• تعریف متغیر weather از نوع string

• با پیام مناسب از کاربر مقداری را برای weather دریافت می کند.

• جلوی عبارت weather is not cloudy خروجی عبارت weather !=cloudy را جایگزین می کند.

که در صورت وارد کردن cloudy پاسخ false و در غیر اینصورت True خواهد بود.

```
weather:
cloudy
weather is not cloudy: False
```

```
weather:
sunny
weather is not cloudy: True
```

خروجی برنامه برای پیام های متفاوت

یک چنبره استادی خودشاد شدیم


از خاک درآمدیم و بر باد شدیم

یک چنبره کودکی با استاد شدیم

پایان سخن شوکه مارا چه رسید

با تشکر از توجهتان

 @iranarze

 @iranarze

جهت دانلود رایگان سوالات استخدامی هنرآموز شبکه و نرم افزار رایانه، اینجا بزنید.