جدول ۴ــ۴ــ مثال‌هایی از فرم نماد علمی

فرم نماد علمی	فرم معمولی
$4\ 38\times 10^{6}$	4380000
$2\ 65\times 10^{-5}$	0000265
$4\ 79832\times 10^{1}$	47 9832
1×10^{7}	1000000
$-5\ 6\times 10^{3}$	−5600

۱ــ۷ــ۴ــ فرم نقطه شناور : در زبان C# از یک فرم نماد علمی‌برای نمایش اعداد اعشاری استفاده می‌شود که به آن **فرم نقطه شناور**[۱] گفته می‌شود. در این فرم مانند نماد علمی، عدد از دو بخش مانتیس و نما تشکیل شده است که با حرف E از یکدیگر جدا شده اند. در این فرم، توان ۱۰ بعد از حرف E نوشته می‌شود وخبری از علامت ضرب بین دو قسمت نیست (جدول ۵ــ۴).

جدول ۵ــ۴ــ مثال‌هایی از نمایش اعداد در فرم نقطه شناور

نمایش عدد در فرم نقطه شناور	عدد
7 5924E1	75 924
1 8E−1	0 18
4 53E − 5	0 0000453
−1 482E0	−1 482
7 8E3	7800 0

بار الکتریکی یک الکترون $۱/۶۰۲ \times ۱۰^{-۱۹}$ کولن است که در متغیرهای زیر ذخیره شده است می‌توانید این عدد بسیار کوچک را در یک متغیر نوع double یا floatبه صورت نقطه شناور ذخیره کنید

```
Double electricalCharge    1.602E   19;
Float electricalCharges    1.602E   19F;
```

سؤال: کدامیک از دستورات بالارا ترجیح می‌دهید؟ چرا؟

۱ــ F oat ng po nt notat on

۲ــ۷ــ۴ــ دقت اعداد قابل نمایش در فرم نقطه شناور : حداکثر تعداد ارقام غیر صفر و با معنی مانتیس عدد را، دقت عدد می‌نامند. دقت اعداد نوع float ۶ یا ۷ رقم و اعداد نوع double ۱۵ رقم است.

نکته

به غیر از میزان حافظه مصرفی و محدوده اعداد قابل نمایش در نوع داده‌های float و dou ble، میزان دقت این دو نوع داده نیز با یکدیگر متفاوت است.

۸ــ۴ــ نوع داده منطقی یا بولین[۱] (bool)

در انتهای جدول ۱ــ۴ نوع داده منطقی یا بولین (bool) را مشاهده می‌کنید این نوع داده فقط شامل دو مقدار درست (true) و نادرست (false) است. متغیرهایی که از این نوع داده تعریف و ایجاد می‌شوند، قادرند یکی از دو مقدار true و false را بپذیرند که با حروف کوچک انگلیسی نوشته می‌شوند.

دستورات زیر متغیر response را اعلان و با false مقدار دهیِ اولیه می‌کند. سپس محتوای متغیر بر روی صفحه نمایش چاپ می‌شود.

```
bool response    false;
System.Console.WriteLine(response );
```

۹ــ۴ــ نوع داده حرفی یا کاراکتری char

کاراکترعبارت است از یک حرف الفباء یا یک علامت و یا نشانه‌هایی مانند آنچه که در روی دکمه‌های صفحه کلید مشاهده می‌کنید. در کامپیوتر برای هر دکمه صفحه کلید یک کدِ عددی در نظر گرفته می‌شود و در واقع هنگامی‌که یک کلید را فشار می‌دهید کدی متناظر با آن کلید تولید و این کد به صورت دنباله‌ای از صفر و یک در حافظه کامپیوتر ذخیره می‌شود. یک کاراکتر را می‌توانید با کد آن مشخص کنید و یا علامت آن را در بین علائم ' ' (تک کوتیشن) قرار دهید.

چند نمونه از کاراکترها را در زیر مشاهده می‌کنید.

```
'A'   ,   'a'   ,   '&'   ,   '$'   ,   '+'   ,   ' '
```

۱ــ Boo ean

نکته

- در داخل علامت‌ها فاصله (Space) نیز به عنوان یک کاراکتر در نظر گرفته می‌شود.
- در داده کاراکتری، فقط یک کاراکتر باید بین علائم ' 'وجود داشته باشد.

توجه داشته باشید که در زبان برنامه‌نویسی C#، نوع داده char به منظور کار با داده‌های کاراکتری پیش‌بینی شده است. اگر بخواهید یک کاراکتر را در یک متغیر ذخیره کنید باید متغیری از نوع داده char تعریف کنید.

گنجایش این متغیر، دو بایت است و کد کاراکتر را نگهداری می‌کند.

```
char نام متغیر ;
```

در دستور زیر، متغیری به نام ch از نوع char تعریف و حرف A در آن ذخیره شده است.

```
char ch  'A';
```

متغیر ch یک متغیر دو بایتی است که در آن کد کاراکتر نگهداری می‌شود. این کد دو بایتی طبق استاندارد یونیکد (Unicode) است. در استاندارد یونیکد، کد هر کاراکترعددی بین ۰ تا ۶۵۵۳۵ است و تمام نشانه‌ها، علائم و حروف الفباء زبان‌های مختلف کشورها به وسیلهٔ این استاندارد کدبندی شده است. این کدبندی مستقل از سیستم عامل، زبان برنامه‌نویسی و سخت افزار است.

در برنامه می‌توانید به جای قرار دادن کاراکتر در علائم ' ' از کد آنها استفاده کنید، چون کاراکترها فقط محدود به آنچه که بر روی صفحه کلید قرار دارد نیستند. بنابراین با دانستن کد هر کاراکتر می‌توانید آن را در برنامه استفاده کنید. معمولاً برای سادگی، این کد را در مبنای ۱۶ ذکر می‌کنند. با توجه به اینکه در کدبندی یونیکد، از دو بایت استفاده می‌شود و هر ۴ بیت یک رقم مبنای ۱۶ است، برای نمایش این کد در مبنای ۱۶ از یک عدد ۴ رقمی استفاده می‌شود. مثلاً کد کاراکتر A عدد ۶۵ در مبنای ۱۰ است. معادل این کد در مبنای ۱۶ عدد ۴۱ است. این عدد را در داخل علائم ' ' قرار می‌دهیم و برای مشخص کردن این عدد به عنوان کد کاراکتر، قبل از آن ، علامت \u یا \x را می‌نویسیم مانند الگوی زیر :

```
'\u کدچهاررقمی'
```

در دستور زیر، متغیر ch اعلان و حرف A در آن ذخیره می‌شود. از صفرهای اضافی قبل از عدد، برای تکمیل کد به صورت ۴ رقمی استفاده شده است.

```
char ch  '\u0041';  // Same as Char ch  'A'
```

۱۰ــ۴ــ نوع داده رشته‌ای (String)

نوع داده char، تنها برای نگهداری یک کاراکتر مناسب است. برای هنگامی که داده‌ها، مانند نام یک شخص، بیش از یک کاراکتر است باید از نوع داده رشته‌ای (string) استفاده کنیم.

یک رشته شامل تعدادی حروف و کاراکتر است که در بین جفت کوتیشن " " قرار گرفته است. مثلاً "Mohammad" یک دادهٔ رشته‌ای شامل ۸ کراکتر است.

۱ــ۱۰ــ۴ــ متغیر رشته‌ای: برای نگهداری داده‌های رشته‌ای در برنامه، باید متغیر رشته‌ای تعریف کنید. متغیرهای رشته‌ای قادرند آدرس محلی که یک داده رشته‌ای وجود دارد را نگهداری کنند یا به عبارت ساده، قادرند داده‌های رشته‌ای را ذخیره کنند. بنابراین با متغیری از نوع رشته، قادر خواهیم بود به داده‌های رشته ای دسترسی داشته باشیم.

دستور زیر یک متغیر رشته‌ای به نام name را اعلان می‌کند.

```
string name;
```

و با دستور انتساب زیر، می‌توانید رشته "Mohammad" را در متغیرname ذخیره کنید و در طول برنامه به آن دسترسی داشته باشید :

```
name  "Mohammad";
```

سؤال: آیا می‌توانید دو دستور بالا را با یک دستور جایگزین کنید؟

۲ــ۱۰ــ۴ــ عملیات بر روی داده‌ها یا متغیرهای رشته‌ای: عملیات مختلفی بر روی رشته‌ها می‌توان انجام داد، یکی از عملیات معمول و کاربردی، الحاق یا کنارهم قرار دادن رشته‌ها است. برای الحاق دو رشته از علامت استفاده می‌شود.

قطعه کد زیر را در نظر بگیرید. در این کدها، محتوای متغیر رشته‌ای name با رشته "Welcome" الحاق شده و حاصل در متغیرmessage قرار می‌گیرد.

```
string name  "Mohammad";
string message  "Welcome"  name;
System.Console.WriteLine(message);
```

نتیجه خروجی چنین خواهد بود :

```
WelcomeMohammad
```

سؤال: اگر بخواهید خروجی به صورت خوانا Welcome Mohammad شود یعنی بین دو کلمه یک فاصله قرار گیرد، چه تغییری در دستورات بالا ایجاد می‌کنید؟

نکته

با توجه به این که در زبان C# ، علامت هم برای عمل جمع ریاضی و هم برای الحاق رشته‌ها استفاده می‌شود، در به کارگیری این علامت در برنامه باید دقت کافی داشته باشید.

کار در کارگاه ۱

مثال ۲ــ۴ : برنامه زیر مانند برنامه ۱ــ۴ برای محاسبه مجموع دو عدد a و b نوشته شده است با این تفاوت که حاصل جمع در متغیری ذخیره نشده بلکه روی صفحه نمایش، نشان داده می‌شود. به خط آخر این برنامه توجه کنید. آیا به نظر شما با اجرای این برنامه، عدد ۲۵ به عنوان حاصل جمع نشان داده می‌شود؟ چرا؟

```
class VariableDemo
{
    static void Main()
    {
        // Declaretwo integer variables
        int a , b ;

          a    10 ;
          b    15 ;

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

        // What is displayed?
        Console.WriteLine("a   b   "   a   b);

    }
}
```

برنامه ۲ــ۴ــ دقت در استفاده از علامت در هنگام کار با اعداد و رشته‌ها

در خط آخر برنامه ۲ــ۴، علامت دو بار استفاده شده است که هر دو علامت، عمل الحاق رشته را انجام می‌دهند . خروجی این برنامه مطابق شکل ۳ــ۴ است :

شکل ۳ــ۴ــ خروجی برنامه ۲ــ۴

برای رفع اشکال برنامه ۲ــ۴، خط آخر را به صورت زیر بازنویسی می کنیم :

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

با تصحیح خط آخر، نتیجه اجرای برنامه شکل ۴ــ۴ خواهد شد :

شکل ۴ــ۴ــ خروجی برنامه۲ــ۴ پس از تصحیح

سؤال: با توجه به خروجی برنامه ۲ــ۴ چرا پرانتز سبب تغییر مقدار خروجی شد؟

۱۱ــ۴ــ دریافت رشته

تاکنون داده‌های مشخص و ثابتی رادر داخل برنامه استفاده کردیم. این داده‌ها به وسیلهٔ برنامه‌نویس درون برنامه تعیین شده بود. حال می‌خواهیم برنامه‌های خود را کاربردی کنیم و داده‌ای را از کاربر دریافت کنیم. برای این منظور از متد ReadLine() استفاده می‌کنیم که به کاربر اجازه می‌دهد تا داده مورد نظر خود را از طریق صفحه کلید وارد کند.

متد ReadLine() مانند متدهایی که تاکنون خوانده ایم در کلاس Console تعریف شده است و در فضای نامی System قرار دارد. بنابراین به‌صورت زیر استفاده می‌شود :

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

کامپیوتر با اجرای این متد متوقف شده و منتظر دریافت داده می‌شود. کاربر می‌تواند داده مورد نظر خود را تایپ کند و در پایان دکمهEnter را بزند که در این صورت، داده به صورت **یک رشته** در حافظه ذخیره می‌شود. اگر رشته دریافتی را با دستور انتساب در یک متغیر رشته‌ای ذخیره کنیم، داده وارد شده، در برنامه قابل دسترسی خواهد بود.

برای مثال می‌خواهیم نام و نام خانوادگی یک شخص را از کاربر سؤال کرده و در برنامه استفاده کنیم. برای این منظور ابتدا دو متغیر رشته‌ای به نام name و family از نوع رشته‌ای اعلان می‌کنیم و سپس از متد ReadLine() برای دریافت نام و نام خانوادگی به صورت زیر استفاده می‌نماییم :

```
string name, family;
name   System.Console.ReadLine();
family   System.Console.ReadLine();
```

نکته

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

توجه داشته باشید وقتی کامپیوتر منتظر دریافت داده است کاربر باید بداند که چه داده‌ای را لازم است وارد کند (نام، نمره، سن) بنابراین لازم است قبل از استفاده از متد ReadLine() یک دستور برای نمایش یک پیام و توضیحی کوتاه در مورد اینکه کامپیوتر منتظر دریافت چه داده‌ای است در برنامه نوشته شود. از متد Write() بدین منظور استفاده می‌کنیم.

مثلاً برای دریافت نام کاربر دستورات زیر را می‌نویسیم :

```
string   name ;
System.Console.Write("Enter your name: ");
name    System.Console.ReadLine();
```

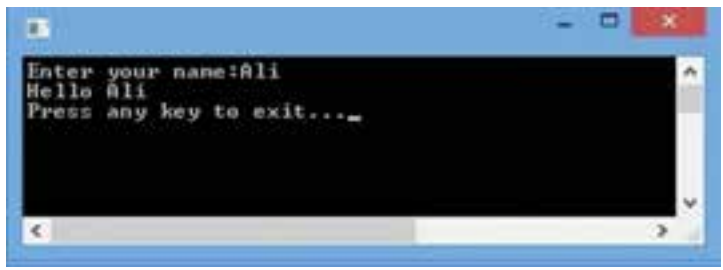
کار در کارگاه ۲

مثال ۳ــ۴ : نام کاربر از ورودی دریافت شده و خطاب به او پیام خوشامدگویی اعلام شود.

```
using System;
class HelloYourName
{
    static void Main()
    {
        string name;
        Console.Write("Enter your name: " );
        name    Console.ReadLine();
        Console.WriteLine("Hello "  name);
        Console.Write("Press any key to exit...");
        Console.ReadKey();
    }
}
```

برنامه ۳ــ۴ــ خوشامدگویی به کاربر

اگر فرض کنید که کاربر، نام Ali را وارد کند، خروجی برنامه به صورت شکل ۵ــ۴ خواهد بود :



شکل ۵ــ۴ــ خروجی برنامه ۳ــ۴

مثال ۴ــ۴ : می‌خواهیم به برنامه ۳ــ۴، دستوراتی اضافه کنیم که علاوه بر دریافت نام کاربر، نام خانوادگی وی نیز سؤال شود و سپس نام ونام خانوادگی را در یک خط نمایش دهد.

در برنامه ۳ــ۴، کافی است یک متغیر رشته‌ای به نام family تعریف کرده و از متد ReadLine() برای دریافت نام خانوادگی استفاده کنیم. برای نمایش نام و نام خانوادگی در یک خط نیز، از علامت برای الحاق رشته‌ها استفاده می‌کنیم (کدهای برجسته، تغییرات جدید هستند).

```
using System;
class HelloYourName
{
    static void Main()
    {
        string name, family;
        Console.Write("Enter your name: ");
        name    Console.ReadLine();

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

        Console.WriteLine("Hello "   name    " "    family");

        Console.Write("Press any key to exit...");
        Console.ReadKey();
    }
}
```

برنامه ۴ــ۴ــ تکمیل برنامه خوشامدگویی به کاربر

سؤال: خروجی برنامه تغییر یافته چه تفاوتی با شکل ۵ــ۴ دارد؟

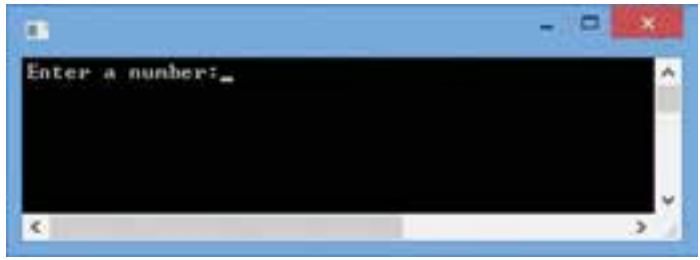
مثال ۵ــ۴ : می‌خواهیم برنامه‌ای بنویسیم که دو عدد دلخواه از کاربر دریافت کند و مجموع آن‌ها را حساب کرده و روی صفحه نمایش، نشان دهد.

برای دریافت داده‌ها از کاربر، از متد ReadLine() مانند مثال‌های قبلی استفاده می‌کنیم. داده‌های دریافتی به وسیلهٔ این متد، در قالب رشته در حافظه ذخیره می‌شوند، بنابراین برای دسترسی به آن ها باید از متغیرهای رشته‌ای استفاده کنیم.

```
using System ;
class GetNumbers
{
    static void Main()
    {
        string firstNumber, secondNumber;

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

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

        Console.WriteLine("Total  "  (firstNumber   secondNumber) );

        Console.Write(“Press any key to exit. . .”);
        Console.ReadKey();
    }
}
```

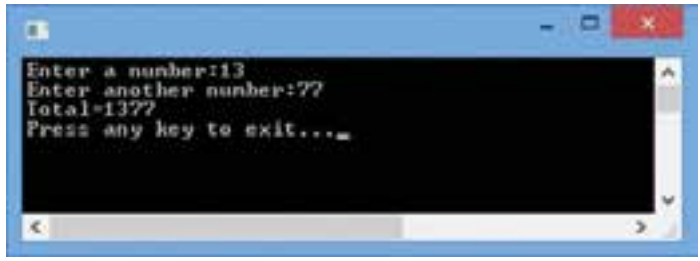
برنامه ۵ــ۴ــ اولین تلاش برای دریافت داده‌های عددی

با اجرای این برنامه، پنجره‌ای ظاهر می‌شود که از کاربر خواسته می‌شود که یک عدد وارد کند (شکل ۶ــ۴).



شکل ۶ــ۴ــ خروجی برنامه ۵ــ۴ دریافت یک عدد

پس از وارد کردن یک عدد و زدن دکمه Enter، عدد دیگری خواسته می‌شود. فرض کنید اعداد ۱۳ و ۷۷ توسط کاربر وارد شود (شکل ۶ــ۴).



شکل ۷ــ۴ــ خروجی برنامه ۵ــ۴

سؤال: کاربر با وارد کردن دو عدد ۱۳ و ۷۷ انتظار داشت که مجموع آنها یعنی عدد ۹۰ روی صفحه نشان داده شود اما به جای آن، عدد ۱۳۷۷ نشان داده شد. چرا؟

همان طور که بیان شد متد ReadLine() داده دریافتی را به صورت یک رشته در حافظه ذخیره می‌کند و در برنامه ۵ــ۴ از متغیرهای رشته‌ای firstNumber و secondNumber برای دسترسی به داده‌های ورودی استفاده کردیم. بنابراین علامت + در دستور زیرعمل الحاق دو رشته مثلاً "13"و "77" را انجام می‌دهد و طبیعی است که نباید انتظار عمل جمع ریاضی داشته باشیم .

۱ــ۱۱ــ۴ــ دریافت اعداد : با توجه به این که داده‌های دریافتی به وسیله متد ReadLine() همواره به صورت رشته تحویل داده می‌شود باید به وسیله دستوری، رشته دریافتی را به عدد تبدیل کنیم. بنابراین به متدی نیاز داریم که بتواند یک رشته شامل ارقام را به ارزش عددی تبدیل کند تا بتوانیم روی آنها محاسبات ریاضی انجام دهیم.

خوشبختانه برای انواع داده‌های عددی، متدی به نام Parse() از قبل تعریف شده است که می‌تواند از یک رشته شامل ارقام، معادل عددی آن را بدست آورد. مثلاً برای تجزیه رشته "259" به ارزش عددی، با توجه به این که درون رشته، یک عدد صحیح قرار دارد، از متد Parse() مربوط به نوع داده int استفاده می‌کنیم :

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

نکته

به عمل بررسی کاراکتر به کاراکتر یک رشته، برای جدا کردن و بدست آوردن یک مقدار با معنی، **تجزیه کردن**[۱] می‌گویند.

حاصل اجرای این متد، عدد ۲۵۹ است که باید در یک متغیر نوع صحیح ذخیره شود. بنابراین استفاده مفید از این متد به صورت زیر خواهد بود :

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

می‌توانید دو دستور بالا را با دستور زیر جایگزین نمایید :

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

اگر رشته‌ای حاوی عدد اعشاری باشد باید از متد Parse() مربوط به نوع داده اعشاری مثلاً float یا double استفاده کنید. مثلاً برای تبدیل رشته "2.50" به عدد 2.5 از دستورات زیر استفاده می‌کنیم :

```
float b ;
b   float.Parse("2.50");
```

با استفاده از متد Parse() می‌توانیم رشته دریافتی که به وسیلهٔ متد ReadLine() از کاربر گرفته شده است را به عدد تبدیل کنیم به شرط اینکه حاوی اعداد باشد.

```
string input;
float number;
input   Console.ReadLine();
number   float.Parse(input);
```

۱ـ Parse

همچنین می‌توانید متد ReadLine() را مستقیماً در متد Parse() استفاده کنید که در این صورت نیازی به متغیر رشته‌ای نیست :

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

سؤال: آیا می‌توانید دو دستور بالا را، باز هم خلاصه‌تر کنید؟

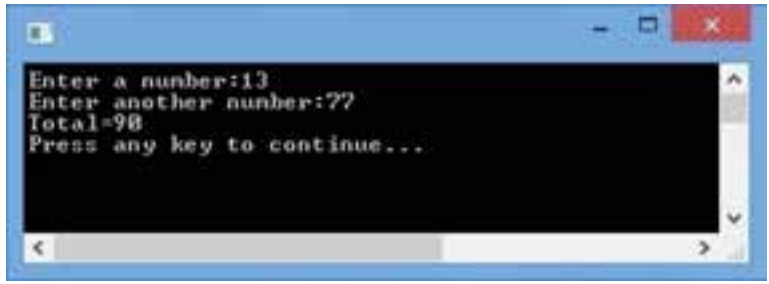
کار در کارگاه ۳

مثال ۶ــ۴ : با تکمیل برنامه ۵ــ۴ مجموع دو عدد دریافتی را چاپ نمایید.

```
using System ;
class GetNumbers
{
    static void Main()
    {
        string input;
        float firstNumber, secondNumber;

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

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

        Console.WriteLine("Total "   (firstNumber   secondNumber));

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

برنامه ۶ــ۴ــ دریافت دو عدد و محاسبه مجموع

نتیجه اجرای برنامه در شکل ۸ــ۴ نشان داده شده است :



شکل ۸ــ۴ــ خروجی برنامه ۶ــ۴

برنامه‌های زیر را در محیط VS ایجاد کنید.

۱ــ دستورات زیر را در داخل متد Main() برای شناسایی انواع متغیرها بنویسید. با اضافه کردن دستورات WriteLine()، محتوای متغیرها را بر روی صفحه نمایش، نشان دهید.

```
// Declare and initialize some variables
// Use long suffix.
long aLongNumber    10000L;

// Use double suffix.
double aDoubleNumber    123.764D;

// Use float suffix.
float aFloatNumber    100.50F;

// Use unsigned suffix.
uint anUnsignedNumber    1000U;

// Use decimal suffix.
decimalaDecimalNumber    4000.1234M;

// Use unsigned suffix and long suffix.
ulonganUnsignedLong    10002000300040005000UL;
```

۲ــ برنامه شماره ۴ــ۴ (خوشامدگویی به کاربر) را با داده‌های مختلف (نام خود، نام همکلاسی‌ها) آزمایش کنید.

۳ــ برنامه شماره ۶ــ۴ (دریافت دو عدد و محاسبه مجموع) را با اعداد صحیح و اعشاری آزمایش کنید.

خودآزمایی فصل چهارم

۱ــ چه نوع حافظه کامپیوتر برای نگهداریِ حجمِ کمی‌از داده‌ها در طول اجرای یک برنامه مناسب است؟

۲ــ در زبان‌های برنامه‌نویسی، مکانی از حافظه برای نگهداری موقتی داده واطلاعات ..... نامیده می‌شود.

۳ــ منظور از نوع داده چیست؟

۴ــ نوع متغیر چه ویژگی‌هایی را نشان می‌دهد؟

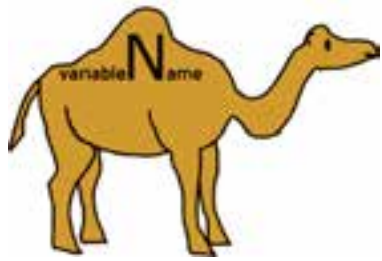
۵ــ تفاوت‌های بین نوع داده float و double را نام ببرید.

۶ــ برای نگهداری هر یک از داده‌های زیر در برنامه، یک متغیر مناسب تعریف کنید.

الف) سن افراد
ب) درجه حرارت محیط اتاق
پ) حقوق کارمند
ت) ظرفیت یک‌هارد دیسک امروزی
ث) جمعیت یک کشور
ج) وضعیت خاموش و روشن بودن یک لامپ

۷ــ تحقیقی کوتاه بر روی روش نام گذاری مجارستانی (Hungarian Notation) با استفاده از اینترنت داشته باشید. با توجه به محیط‌های برنامه‌نویسی (IDE) پیشرفته، مانند ویژوال استودیو، نشان دهید که دیگر نیازی به ذکر نوع داده در ابتدای نام متغیر که در روش مجارستانی استفاده می‌شود، وجود ندارد.

۸ــ شکل زیر چه روشی را برای نام گذاری متغیرها نشان می‌دهد؟ نام این روش چیست؟



۹ــ چرا در هنگام ترجمه دستور زیر، خطا ظاهر می‌شود؟چگونه این خطا را برطرف می‌کنید؟

```
short value   66000 / 2 ;
```

۱۰ــ در جدول زیر، کدامیک از نام‌های متغیر، غیر مجاز است و یا مناسب نیستند. آنها را تصحیح کنید. (اولین ردیف جدول برای شما پاسخ داده شده است.)

نام پیشنهادی شما	نام متغیر			کاربرد متغیربا توجه به معنی آن	نام متغیر
	غیرمجاز	نامناسب	مجاز		
	☐	☐	■	وضعیت شبکه	networkOK
	☐	☐	☐		29yesitsme
	☐	☐	☐		myCurrentID
	☐	☐	☐		intValue
	☐	☐	☐		8٪tax
	☐	☐	☐		woodLength
	☐	☐	☐		glassArea
	☐	☐	☐		width
	☐	☐	☐		height
	☐	☐	☐		MySalary
	☐	☐	☐		employeeSalary

۱۱ــ کدامیک از داده‌های زیر یک داده کاراکتری محسوب می‌شود؟ برای پاسخ خود دلیل بیاورید.

'۰۸' , '۱۲' , '۱' , '&' , '!' , '+' , 'abc' , ' ' , '!'

۱۲ــ در زبان C#، برای دریافت داده از ورودی، از متد ..... و برای نمایش اطلاعات در خروجی، از متد..... استفاده می‌کنیم.

۱۳ــ اعداد زیر را در متغیرهای مناسب با استفاده از روش نقطه شناور جای دهید.

الف) عمر زمین بر حسب سال ۴,۶۰۰,۰۰۰,۰۰۰

ب) فاصله زمین تا خورشید ۱۴۹,۶۰۰,۰۰۰,۰۰۰ متر است.

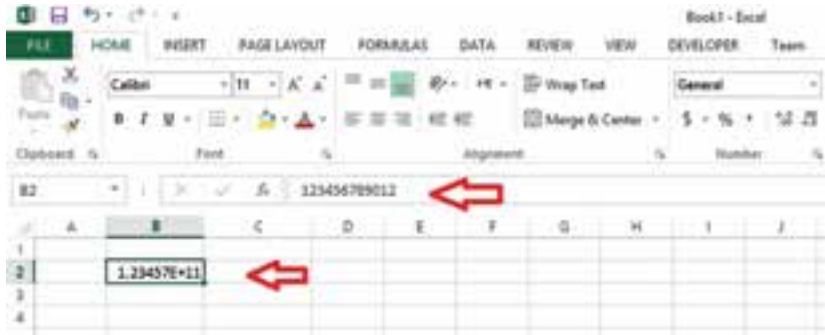
پ) اندازه جرم یک اتم کربن (جرم اتمی) برابر با ۱۶۶۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰ kg

۱۴ــ با توجه به اینکه لازم نیست مانتیس بین ۰ تا ۱ باشد، جدول زیر را تکمیل کنید.

عدد	نمایش عدد در فرم نقطه شناور	گونه دیگری از فرم نقطه شناور
75 924	7 5924E1	
0 18	1 8E−1	
0 0000453	4 53E−5	
−1 482	−1 482E0	
7800 0	7 8E3	

۱۵ــ در برنامه‌های صفحه گسترده، اعداد بزرگ به صورت نقطه شناور نیز نمایش داده می‌شوند. وارد برنامه اکسل شوید و عدد بزرگی را وارد نموده و کلید Enter را بزنید. چه عددی روی صفحه نشان داده می‌شود؟

پس از مشاهده فرم نقطه شناور، سعی کنید با تغییر فرمت سلول مربوطه، عدد را به صورت معمولی نشان دهید.



۱۶ــ کدام‌یک از دستورات زیر،می‌تواند مجموع دو عدد a و b را به‌طور صحیح نشان دهد؟ نتیجه اجرای هر یک از دستورات را نیز بنویسید.

```
int  a  10, b  20;
System.Console.WriteLine("a" + "b");
System.Console.WriteLine("a + b");
System.Console.WriteLine(a + b);
System.Console.WriteLine("a + b"+a + b);
```

۱۷ــ سؤال زیر به زبان انگلیسی است. آن را خوانده و پاسخ صحیح را انتخاب کنید.

The C# method that prints a line of output on the screen and then positions the cursor on the next line is .....

A) println()
C) WriteLine()
B) DisplayLine()
D) Write()

تمرینات برنامه‌نویسی فصل چهارم

۱ــ برنامه‌ای بنویسید که دو عدد از کاربر دریافت نماید و حاصل جمع و حاصل تفریق آنها را نمایش دهد. (اعداد ورودی ممکن است صحیح و یااعشاری باشد.)

۲ــ برنامه‌ای بنویسید که نام و نام خانوادگی و سن کاربر را دریافت کند و سپس اطلاعات دریافتی را با رنگ‌های دلخواه روی صفحه نمایش، نشان دهد.

۳ــ با استفاده از یک برنامه ساده شامل متد ()WriteLine، رشته‌ها یا کاراکترهای جدول زیر را نمایش دهید تا بتوانید معادل آن‌ها را پیدا کرده و جدول را کامل کنید.

کد حرف یا رشته	معادل	کد حرف یا رشته	معادل
'\u0007'		"0\u00200"	
"b\u0061ck"		"C\u0023"	
'u000a\'		'\u0040'	
"12" + "8"		'\u0030'	

واژگان و اصطلاحات انگلیسی فصل چهارم

ردیف	واژه انگلیسی	معنی به فارسی
۱	Assignment	
۲	Boolean	
۳	Built–In Data Type	
۴	Camel Notation	
۵	Concatenate	
۶	Floating point notation	
۷	Hungarian Notation	
۸	Initialize	
۹	Integer Numbers	
۱	Precision	
۱۱	Primitive Data Type	
۱۲	Random Access Memory	
۱۳	Representation	
۱۴	Significant digits	
۱۵	Unsigned numbers	
۱۶	Variable	

فصل پنجم

عبارت‌های محاسباتی

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

پس از پایان این فصل انتظار می‌رود که فراگیر بتواند :

۱ــ عملوند، عملگر و عبارت محاسباتی را تعریف کند و آنها را به درستی به کارگیرد.

۲ــ عملگرهای ریاضی را نام ببرد و در عبارت محاسباتی از آنها استفاده کند.

۳ــ حاصل عبارت محاسباتی را با استفاده از جدول تقدم عملگرها به دست آورد.

۴ــ کاربرد عملگرهای افزایشی، کاهشی و انتساب را بیان نماید.

۱ــ۵ــ عبارت[۱] چیست؟

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

$8 \quad 3*5$

$9-7.25$

$5.6*3 \quad 5.6$

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

۱ــ Express on

در این عبارت‌ها، علامت نشانهٔ عمل جمع و علامت * نشانهٔ عمل ضرب است. به این علامت‌ها که بیانگر انجام یک عمل بر روی اعداد و داده‌ها هستند، **عملگر**[1] گفته می‌شود. مثلاً عملگر * در عبارت 5*3 8 بر روی اعداد 3 و 5 عمل ضرب را انجام می‌دهد و همچنین عملگر در عبارت بالا بر روی عدد 8 و نتیجه حاصلضرب یعنی عدد 15، عمل جمع را انجام می‌دهد. به اعدادی که یک عملگر بر روی آنها عملی را انجام می‌دهد **عملوند**[2] می‌گویند. اعداد 3 و 5 عملوندهای عملگر ضرب و عدد 8 و 15 عملوندهای عملگر جمع هستند.

هر یک از عملگرهای ضرب و جمع بر روی دو عدد عمل می‌کنند و به عبارتی دارای دو عملوند هستند به این عملگرها، عملگرهای **دوتایی**[3] گفته می‌شود. عملگر در عبارت 7.25 9 عملگر تفریق است که آن نیز یک عملگر دوتایی است و حاصل تفریق 7.25 از 9 را محاسبه می‌کند. اما عملگر قرینه در عبارت x ، فقط دارای یک عملوند x است و آن را قرینه می‌کند. این عملگر یک عملگر **یکتایی**[4] است.

نکته

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

۲ــ۵ــ عملگرهای ریاضی یا حسابی[5]

اگر در عبارتی بیش از یک عملگر وجود داشته باشد ابتدا عملگری عمل خود را انجام می‌دهد که اولویت[6] بالاتری نسبت به دیگری داشته باشد. مثلاً اولویت عملگر ضرب بیش از اولویت عملگر جمع است. چنانچه دو یا چند عملگر دوتایی، با اولویت یکسان در یک عبارت وجود داشته باشد ابتدا عملگر سمت چپ انجام می‌شود. به عبارت دیگر از سمت چپ به راست، عملگرها به ترتیب انجام می‌شوند که به آن «**شرکت‌پذیری چپ**[7]» می‌گویند.

در جدول ۱ــ۵ لیست عملگرهای ریاضی را به ترتیب اولویت مشاهده می‌کنید. عملگر قرینه اولویت بالاتری نسبت به بقیه عملگرهای ریاضی دارد و عملگرهای جمع و تفریق دارای اولویت یکسان ولی کمترین اولویت را در بین عملگرهای ریاضی دارند.

۱ــ Operator
۲ــ Operand
۳ــ B nary Operator
۴ــ Una y Operator
۵ــ Ar thmet c
۶ــ Precedence
۷ــ Left_Assoc at ve

جدول ۱ــ۵ــ عملگرهای ریاضی

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

عملکرد عملگرهای جمع و تفریق و ضرب مانندعملکرد آنها در ریاضیات است اما عملگر تقسیم با توجه به نوع عملوندهایش می‌تواند تقسیم صحیح و بدون ممیز و یا تقسیم اعشاری و ممیزی انجام دهد. مثلاً در عبارت 2 / 9 چون عملوندها اعداد صحیح هستند بنابراین تقسیم بدون ممیز و صحیح انجام خواهد شد که نتیجه آن عدد ۴ است . اما در عبارت 9.0 / 2 یا در عبارت 9 / 2.0 ، چون حداقل یکی از عملوندها، اعشاری است بنابراین تقسیم به صورت اعشاری انجام می‌شود که حاصل عبارت عدد ۴/۵ است.

در جدول ۱ــ۵، عملگر جدیدی نیز به نام باقیمانده تقسیم که با نشانهٔ ٪ مشخص می‌شود، مشاهده می‌کنید. به وسیلهٔ این عملگر می‌توانیم باقیماندهٔ تقسیم یک عدد بر عدد دیگر را با توجه به خارج قسمت صحیح و بدون اعشار به دست آوریم. مثلاً در تقسیم عدد ۲۳ بر عدد ۵، خارج قسمت بدون اعشار عدد ۴ است بنابراین باقیمانده عدد ۳ است.

```
23 / 5   4
23 ٪ 5   3
```

```
۲۳ | ۵
۲۰ | ۴
---
 ۳
```

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

```
5.6 *(3   6.5)
```

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

در جدول ۲ــ۵، چند نمونه از عبارت‌های ریاضی نشان داده شده است.

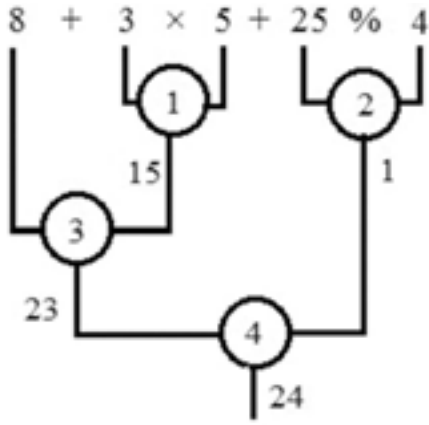
جدول ۲ــ۵ــ مثالی از عبارت‌های ریاضی

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

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

8 3 * 5 25 ٪ 4

در این عبارت بیش از یک عملگر وجود دارد، بنابراین ابتداعملگری که دارای اولویت بالاتر است، انجام می‌شود. چون اولویت عملگرهای * و ٪ بالاتر از عملگر است بنابراین ابتدا این دو عملگر انجام می‌شود و از طرفی چون این دو عملگر دارای اولویت یکسان هستند، برطبق شرکت‌پذیریِ چپ، ابتدا عملگر سمت چپ یعنی * و سپس عملگر ٪ انجام می‌شود. بر همین اساس در مورد عملگرهای جمع نیز ابتدا عملگر سمت چپ و سپس عملگر بعدی انجام می‌شود (نمودار ۱ــ۵).



نمودار ۱ــ۵ــ ترتیب اجرای عملگرها

در برنامه‌ها، معمولا حاصل یا نتیجهٔ یک عبارت را در یک متغیر نگهداری می‌کنند. البته نوع متغیری که حاصل یک عبارت، در آن قرار می‌گیرد باید با نوع عبارت، سازگار باشد. مانند ظرفی در آشپزخانه که بخواهیم در آن غذا یا نوشیدنی بریزیم باید گنجایش مناسب آن غذا را داشته باشد.

قوانین زیر، باید به وسیلهٔ برنامه نویس، در هنگام انتساب یک عبارت به یک متغیر، رعایت شود، در غیر اینصورت با پیام خطای مترجم مواجه می‌شویم. مترجم زبان C# روی این قوانین سخت گیر[1] است زیرا می‌خواهد از اشتباهات برنامه نویسان جلوگیری نماید، این یکی از ویژگی‌های زبان C# است.

۱ــ اگر حاصل یک عبارت عدد صحیح باشد بسته به اندازه و بزرگی عدد، می‌تواند در یک متغیرِ نوعِ صحیح که گنجایش آن مساوی یا بزرگ‌تر از حاصل عبارت باشد جای گیرد.

مثلاً حاصل عبارت 31 / 175 عدد 5 است این عدد کوچک می‌تواند در تمام متغیرهای نوع صحیح زیر قرار گیرد.

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

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

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

مثلاً در دستور زیر حاصل عبارت 1234567890 / 10 در متغیر اعشاری ذخیره می‌شود ولی به دلیل اینکه عدد بزرگی است به صورت گرد شده در متغیر ذخیره می‌شود (شکل ۲ــ۵).

```
float number    1234567890 /  10;
Console.WriteLine("Number  "    number);
```

شکل ۲ــ۵ــ نتیجه اجرای عملگر تقسیم

۱ــ Str ct type check ng anguage