فصل چهارم

آشنایی با انواع داده‌ها و متغیرها

برنامه‌هایی که تاکنون نوشته‌ایم، به نشان دادن یک پیام یا حاصل یک عبارت بر روی صفحه نمایش محدود می‌شد، اما در برنامه‌های کاربردی با داده‌ها و مقادیر مختلف سروکار داریم و باید بر روی آن عملیاتی را انجام دهیم. بعضی از این مقادیر مانند تاریخ تولد یک شخص یا نمره یک دانش‌آموز از قبل مشخص نیستند. مقدار این نوع داده‌ها باید در هنگام اجرای برنامه، ابتدا از کاربر دریافت شوند و در مکانی از حافظه کامپیوتر نگهداری شوند و در ادامه برنامه و در جریان پردازش، مورد استفاده قرار گیرند. چه حافظه‌ای برای نگهداری داده‌ها در هنگام پردازش مناسب است؟

در این فصل با متغیرها آشنا می‌شویم که برای نگهداری موقتی داده‌ها در برنامه مورد استفاده قرار می‌گیرند. همچنین برای نگهداری اطلاعات و نمایش آنها بر روی صفحه نمایش از متغیرها استفاده می‌کنیم.

پس از پایان این فصل انتظار می‌رود که فراگیر بتواند :

۱ــ متغیر را تعریف کند و انواع متغیر را در برنامه‌های خود به کاربندد.
۲ــ انواع داده‌ها را نام ببرد و تفاوت کاربرد هر یک را توضیح دهد.
۳ــ میزان حافظه و محدودهٔ انواع داده‌ها را بیان کند.
۴ــ متغیرها را به‌طور صحیح در برنامه اعلان کندو آن‌ها را مقداردهی نماید.
۵ــ شکل نمایش نقطه شناور را توضیح دهد و اعداد اعشاری را در این قالب بنویسد.
۶ــ ازمتد ()ReadLine برای دریافت داده‌های یک برنامه از ورودی استفاده کند.
۷ــ بر روی رشته دریافتی از ورودی، تغییراتی داده و سپس نمایش دهد.
۸ــ از متد ()Parse برای تبدیل یک رشته به یک عدد استفاده کند.

۱ــ۴ــ متغیر چیست؟

در هر کامپیوتر حافظه‌های مختلفی وجود دارد که هر یک برای انجام کار خاصی پیش‌بینی شده است. یک نوع از حافظهٔ کامپیوتر که قادر است داده‌ها را نگهداری کند و به سرعت قابل دسترسی است، حافظه موقتی RAM[۱] است. از اطلاعات درون حافظه RAM، در هر لحظه می‌توان با اطلاع شد و یا در صورت لزوم محتویات آن را تغییر داد یا مقدار جدیدی را در آن ذخیره کرد.

با توجه به مطالب گفته شده، لازم است در یک برنامه، یک یا چند مکان (بایت) از حافظه RAM کامپیوتر برای نگهداری موقتی داده‌ها یا نتایج حاصل از پردازش مورد استفاده قرار گیرد. در زبان‌های برنامه‌نویسی به این مکان‌ها، متغیر[۲] گفته می‌شود زیرا می‌توان محتوای آنها را در طول اجرای برنامه تغییر داد.

نکته

متغیر : مکانی از حافظه RAM کامپیوتر است که برای نگهداری موقتی داده‌ها یا اطلاعات استفاده می‌شود

متغیر را مانند یک ظرف در نظر بگیرید. در آشپزخانه ظروف متعددی با شکل ظاهری، اندازه و جنس مختلفی وجود دارد که هر یک برای نگهداری یک نوع غذا یا مایعات استفاده می‌شود که گنجایش آن را داشته باشد. در یک برنامه نیز برای نگهداری هر یک از داده‌ها با توجه به نوع و بزرگی داده، باید از متغیر مناسبی استفاده کنیم که بتواند داده را نگهداری کند.

۲ــ۴ــ روش اعلان (تعریف) و ایجاد متغیرها

قبل از اینکه بتوانید مقداری را در یک متغیر ذخیره کنید باید متغیری را ایجاد کنید که قادر باشد آن مقدار را به درستی ذخیره نماید. در هنگام ایجاد متغیر باید نوع متغیر را مشخص نمایید. در زبان C# برای ایجاد و مشخص کردن نوع متغیر، از الگوی زیر استفاده می‌شود.

نام متغیر نوع داده;

دستور زیر را در نظر بگیرید :

```
int a;
```

۱ــ Random Access Memory

۲ــ Var ab e

در این دستورمتغیر a از نوع عدد صحیح اعلان می‌شود. کلمه int نوع متغیر را مشخص می‌کند که قادر است اعداد صحیح را در خود نگهداری کند و a نام متغیر است. نام متغیر به وسیلهٔ برنامه نویس انتخاب می‌شود که بهتر است نام و نوعِ آن مطابق با داده‌ای باشد که مقداردهی می‌شود.

۳ــ۴ــ نوع داده[۱] (نوع متغیر)

نوع متغیربه طور کلی ۳ ویژگی را مشخص می‌کند :

۱ــ گنجایش یا ظرفیت متغیر : مثلاً نوع int چهار بایت است.

۲ــ نوع اطلاعاتی که در متغیر می‌توان ذخیره کرد : مثلاً در متغیر نوع int فقط اعداد صحیح و بدون ممیز قابل نگهداری است.

۳ــ چه عملیاتی را می‌توان بر روی آن انجام داد : مثلاً عملیات ریاضی معمول را می‌توان روی اعداد نوع int انجام داد.

در زبان C# علاوه بر نوع داده int، انواع دیگری از داده‌ها نیز دسته بندی و گروه بندی شده‌اند و نحوه نمایش یا نگهداری[۲] آنها در حافظه و عملیاتی که می‌توان بر روی آنها انجام داد از قبل مشخص و تعریف شده است و برای هر دسته یا گروه از داده‌ها، یک نام انتخاب شده است که به آن نوع داده اولیه[۳] یا درون ساخته می‌گویند. جدول ۱ــ۴ انواع داده و مشخصات هر یک را نشان می‌دهد.

برای مثال در جدول ۱ــ۴ نوع داده sbyte را در نظر بگیرید. این نوع داده، اعداد صحیح و بدون ممیز در محدوده ۱۲۸ تا ۱۲۷ را شامل می‌شود که یک بایت حافظه را اشغال می‌کند و بر روی آنها می‌توان عملیات ریاضی را انجام داد. اگر در یک برنامه، متغیری از نوع sbyte را استفاده کنیم، قادر خواهیم بود به عنوان مثال عدد ۷۸ را در آن ذخیره کنیم. اما نمی‌توان عدد ۲۰۰ و یا عدد ۱/۵ را در آن نگهداری کرد. همچنین نوع داده byte اعداد صحیحی فقط در محدوده ۰ تا ۲۵۵ را شامل می‌شود که در یک بایت قرار می‌گیرد. در این نوع داده فقط اعداد مثبت یا بدون علامت[۴] قابل نمایش می‌باشند.

---

۱ ــ Data Type

۲ ــ Representat on

۳ ــ Pr m t ve Data Type or Bu t In Data Type

۴ ــ Uns gnex numbers

جدول ۱ــ۴

	نوع داده	کاربرد نوع داده	مقدار حافظه (بایت)	کمترین مقدار	بیشترین مقدار
عددی	sbyte	اعداد صحیح		28	27
	byte	اعداد صحیح مثبت		0	255
	sho t	اعداد صحیح	2	32768	32767
	usho t	اعداد صحیح مثبت	2	0	65535
	nt	اعداد صحیح	4	2 47483648	2 47483647
	u nt	اعداد صحیح مثبت	4	0	4294967295
	ong	اعداد صحیح	8	9223372036854778508	9223372036854778507
	u ong	اعداد صحیح مثبت	8	0	844674407370955 6 5
	f oat	اعداد اعشاری	4	3.402823×0^{38}	3.402823×0^{38}
	doub e	اعداد اعشاری با دقت زیاد	8	$.797693\ 3486232 \times 0^{308}$	$.797693\ 3486232 \times 0^{308}$
	dec ma	اعداد صحیح بزرگ اعداد اعشاری با دقت بسیار زیاد	6	79228162514264337593543950335 7.9×0^{28}	79228162514264337593543950335 7.9×0^{28}
غیرعددی	boo	مقدار منطقی		fa se	t ue
	char	یک حرف یا علامت (کراکتر)	2	0 کد کراکتر مطابق با سیستم Un code	65535 کد کراکتر مطابق با سیستم Un code
	str ng	رشته		–	–
	object	آدرس یک داده		–	–

دستور byte age ; متغیری به نام age ایجاد می‌کند که این متغیر بسیار کوچک و به ظرفیت یک بایت است و می‌تواند یکی از اعداد صفر تا ۲۵۵ را در خود ذخیره کند.

اگر بخواهید چند متغیر از یک نوع را تعریف کنید کافی است بعد از ذکر نوع داده، نام متغیرها را با علامت ویرگول از یکدیگر جدا کنید. مثلاً برای تعریف دو متغیر برای نگهداری حداقل و حداکثر درجه حرارت از دستور زیر استفاده می‌کنیم :

```
Sbyte minTemp , maxTemp ;
```

نکته

هر نوع داده، مجموعه‌ای از مقادیر به همراه مجموعه‌ای از عملیات را مشخص می‌کند.

برای اعداد صحیح و بدون ممیز نوع داده‌های زیر استفاده می‌شود :

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

و برای اعداد اعشاری می‌توانید از نوع داده‌های float و double استفاده کنید. نوع داده float برای اعداد اعشاری با دقت حداکثر ۷ رقم اعشار استفاده می‌شود. در صورتی که ارقام عدد بیش از آن باشد عدد گرد می‌شود. مثلاً عدد ۱۲۳/۴۵۶۷۸۹ به صورت عدد ۱۲۳/۴۵۶۸ قابل نگهداری است. نوع دادهdouble برای اعداد اعشاری بسیار بزرگ و یا بسیار کوچک مانند جرم و بار الکتریکی یک الکترون و با دقت زیاد ۱۵ رقم استفاده می‌شود.

نکته

در زبان برنامه‌نویسی C#، قبل از اینکه بتوانید داده‌ای را در یک متغیر ذخیره کنید باید متغیر را ایجاد (یا اعلان) کنید و در هنگام ایجاد کردن یک متغیر، باید نوع متغیر(نوع داده) را مشخص نمایید. مثال float mark;

۴ــ۴ــ مقداردهی متغیرها

پس از تعریف یا ایجاد متغیر، می‌توانید در آن، مقداری را با توجه به نوع متغیر ذخیره کنید. توجه داشته باشید که **در یک متغیر همواره فقط یک مقدار نگهداری می‌شود و با ذخیره کردن داده جدید در یک متغیر، مقدار قبلی آن از بین می‌رود.** مقداردهی متغیرها به چند روش صورت می‌گیرد. با دستور زیر مستقیما مقداری در متغیر قرار می‌گیرد به این دستور، دستور انتساب[۱] می‌گویند.

نام متغیر = مقدار;

دستورات زیر را در نظر بگیرید :

```
byte  age ;
age    16 ;
```

متغیر age از نوع عدد صحیح اعلان شده و با عدد ۱۶ مقداردهی شده است.

۱ــ Ass gnment

در هنگام تعریف یا ایجاد متغیر نیز می‌توانید آن را مستقیما مقداردهی کنید که به آن **مقداردهی اولیه**[1] می‌گویند. الگوی آن چنین است :

نام‌متغیر نوع‌داده = مقدار;

بنابراین دو دستور قبل را با الگوی بالا جایگزین می‌کنیم :

```
byte  age  16 ;
```

نکته

۱ــ برای مشخص کردن اعداد مثبت نیازی به قراردادن علامت در پشت عدد نیست.

۲ــ در بین ارقام عدد نباید ویرگول قرار دهید تا ارقام عدد، دسته بندی و جدا شوند.

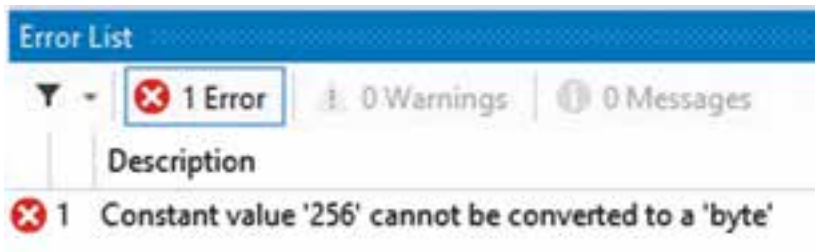
۳ــ اگر عددی را بخواهید در داخل یک متغیرذخیره کنید که خارج از ظرفیت و گنجایش آن متغیر باشد، مترجم متوجه آن شده و اجازه نمی‌دهد.

مثلاً دستور انتساب زیر را در نظر بگیرید :

```
byte  age  256 ;
```

با توجه به ظرفیت متغیر age که حداکثر عدد ۲۵۵ است، در هنگام ترجمه این دستور، خطای شکل ۱ــ۴ ظاهر می‌شود که شرح آن چنین است :

«مقدار ثابت ۲۵۶ را نمی‌تواند به یک byte تبدیل شود».



شکل ۱ــ۴ــ خطا در انتساب عدد صحیح ۲۵۶ در یک متغیر نوع byte

۱ــ In t a ze

در یک برنامه به زبان C# می‌توانید اعداد صحیح را در مبنای ۱۶ نیز بنویسید. برای این منظور قبل از عدد مورد نظر از پیشوند 0x یا 0X استفاده کنید که نشانه اعداد مبنای ۱۶ می‌باشد. مثلاً :

```
byte portValue    0x1B;
ushort portAddress    0X00FF;
```

با اجرای این دستورات در متغیر portValue عدد ۲۷ و در متغیر portAddress عدد ۲۵۵ قرار می‌گیرد[۱].

برای مشخص کردن انواع عددی دیگر از نشانه‌های جدول ۲ــ۴ استفاده می‌شود که در انتهای عدد ذکر می‌شود.

جدول ۲ــ۴ــ نشانه‌های نوع اعداد ثابت

نوع عدد	نشانه	مثال
عدد صحیح مثبت	u یا U	125U
عدد صحیح بزرگ	l یا L	1700L
عدد صحیح بزرگ مثبت	UL	250000UL
عدد اعشاری با دقت معمولی	f یا F	2 5f
عدد اعشاری با دقت زیاد	d یا D	12 75d
عدد بسیار بزرگ	m یا M	12345678M

نکته

اگر در برنامه، یک عدد اعشاری بدون نشانه بنویسید این عدد به عنوان عدداعشاری با دقت زیاد در نظر گرفته می‌شود.

برای ذخیره اعداد اعشاری باید ازمتغیرهای نوع float یا double استفاده کنید. مثلاً برای نگهداری نمرات درسی(معمولاً با دو رقم اعشار) یا اعداد گنگ مانند π باید از چنین متغیرهایی استفاده کرد.

دستور زیر را در نظر بگیرید :

```
double PI    3.141592653589793238;
```

در این دستور برای نگهداری عدد π متغیر PI با دقت زیاد اعلان و مقداردهی شده است.

---

۱ . (1B) = (1×16)+11=27 (FF)=(15×16) + 15=255

نکته

برای ذخیره اغلب داده‌ها مانند نمره یک درس، متغیر نوع float مناسب است. اگر چه می‌توانید از متغیر نوع double نیز استفاده کنید ولی حافظهٔ اشغالی این متغیر دو برابر متغیر نوع float است.

دستورات زیر را در نظر بگیرید :

```
float myPhysicMark;
myPhysicMark   17.75f;
```

در دستورات بالا برای ذخیرهٔ نمرهٔ درس فیزیک متغیری اعلان و مقداردهی شده است.

سؤال: در دستور انتساب، بعد از عدد اعشاری 17.75 حرف f نوشته شده است که نشانه اعداد اعشاری با دقت معمولی است. آیا می‌توانید حرف f را ننویسید؟

نکته

در زبان C# هر عدد اعشاری داخل برنامه، به وسیلهٔ مترجم به عنوان نوع double در نظر گرفته می‌شود. بنابراین اگر بخواهید یک عدد ممیزی را در یک متغیر نوع float ذخیره کنید مترجم خطا یا هشدار می‌دهد. برای جلوگیری از این مسئله باید از متغیرهای نوع double در هنگام کار با اعداد اعشاری استفاده کنید و یا اینکه در جلوی اعداد اعشاری حرف F یا f را بنویسید تا مترجم، این عدد را به عنوان یک عدد نوع float در نظر بگیرد.

۵ــ۴ــ نشان دادن محتوای متغیرها بر روی صفحه نمایش

معمولاً در برنامه‌ها لازم است محتوای متغیرها که شامل داده‌ها و یا نتایج پردازش با اطلاعات بر روی صفحه نشان داده شود تا کاربر از آنها آگاه شود. بدین منظور از متد Write() یا WriteLine() استفاده می‌کنیم که در فصل‌های قبلی برای نمایش یک پیام یا حاصل یک عبارت به کار گرفته شد. مثلاً برای نشان دادن محتوای متغیر age دستور زیر را می‌نویسیم :

```
byte  age   16 ;
System.Console.WriteLine( age );
```

با توجه به اینکه در متغیر age عدد ۱۶ قرار دارد با اجرای دستور بالا، این عدد روی صفحه کنسول نشان داده می‌شود.

اگر شخص دیگری غیر از شما، این عدد را روی صفحه مشاهده کند، شاید متوجه نشود که این عدد چیست و شاید عدد ۱۶ را به عنوان نمره در نظر بگیرد. بنابراین بهتر است قبل از نمایش هرعدد، یک پیام (رشته) نیز نشان داده شود و به صورت کوتاه و مختصر منظور و مفهوم عددی را که قرار است روی صفحه نشان داده شود بیان کند. بنابراین دستور بالا را به صورت زیر می‌نویسیم :

```
System.Console.WriteLine("My age is "  age );
```

با اجرای این دستور، عبارت زیر روی صفحه نشان داده می‌شود :

```
My age is 16
```

علامت در دستور بالا، به معنای عمل جمع ریاضی نیست بلکه به منظور کنار هم قرار دادن[1] این دو مقدار (رشته‌ها) استفاده شده است. همان طور که در دستور زیر نیز از علامت استفاده شده است :

```
System.Console.WriteLine("I am "  age  "years old." );
```

با اجرای این دستور، عبارت زیر روی صفحه نشان داده می‌شود :

```
I am 16 years old.
```

مثال ۱ــ۴ــ استفاده از چند متغیر صحیح و اعشاری در برنامه ۱ــ۴ ، نشان داده شده است :

```
class VariableDemo
{
   static void Main()
    {
       // Declare some integer numbers variables
       int a   10 , b   20 ,c ;
       c   a   b;
       Console.WriteLine("a  "   a);
       Console.WriteLine("b  "   b );
       Console.WriteLine("a  b  "   c);
       // Declare some real numbers variables
       float lowPI   3.14159265358979238f;
       double highPI   3.141592653589793238;
```

۱ــ Concatenate

```
            // Print the results on the console
            Console.WriteLine("Float PI is:  "   lowPI);
            Console.WriteLine("Double PI is:  "   highPI);

            Console.ReadKey();
        }
    }
```

برنامه ۱ــ۴ــ تعریف و مقداردهی و نمایش محتوای متغیرها

در برنامه ۱ــ۴، سه متغیر a , b , c از نوع عدد صحیح تعریف شده‌اند و در متغیر c نتیجه حاصل جمع دو عدد a و b قرار می‌گیرد. در دو متغیر اعشاری lowPI و highPI عدد π با دقت‌های مختلف نگهداری شده است.

شکل ۲ــ۴ــ خروجی برنامه ۱ــ۴

۶ــ۴ــ نحوه نام گذاری متغیرها

همان طور که پدر و مادر برای انتخاب یک نام خوب و مناسب برای فرزند خود، وقت زیادی می‌گذارند و نکاتی از جمله زیبایی نام و با معنا بودن را رعایت می‌کنند و همچنین سعی می‌کنند که این نام قبلاً در خانواده و یا نزدیکان انتخاب نشده باشد، به همان صورت برنامه نویس نیز برای متغیرها باید یک نام صحیح، بامعنا و غیرتکراری در محدوده آن را انتخاب کند این کار باید با حوصله انجام شود و نام انتخابی نباید با نام‌های دیگر یکسان باشد.

در زبان C# در نام گذاری متغیرها، رعایت موارد زیر الزامی است :

۱ــ استفاده از حروف الفبا، اعداد و کاراکتر زیرخط، مجاز است.

۲ــ نام متغیر نمی‌تواند با عدد شروع شود.

۳ــ نام انتخابی نمی‌تواند با کلمات کلیدی یا رزرو شده باشد.

۴ــ استفاده ازعلامت فاصله و خط تیره در نام متغیر مجاز نیست.

در انتخاب نام متغیرها، بهتر است نکات زیر رعایت شود :

- نام با معنی و با توجه به کاربرد متغیر در برنامه انتخاب شود. مانند woodLength
- از نام‌های مخفف استفاده نکنید چون خواندن آنها مشکل است. مانند crntStdnt
- اولین حرف نام متغیر را با حروف کوچک شروع کنید و اگر نام متغیر از چند کلمه تشکیل شده، برای خوانایی، حرف اول کلمات بعدی را با حروف بزرگ بنویسید. به این روشِ نوشتن نام، کوهان شتری[۱] می‌گویند.

چند نمونه نام متغیر دو کلمه‌ای به روش کوهان شتری را ملاحظه می‌کنید :

fileName , userName , notFound , localIP

روش دیگری برای نام‌گذاری متغیرها به‌نام روش مجارستانی[۲] به وسیلهٔ آقای چارلز سیمونیی[۳] ابداع شده که در ابتدای نام متغیر، مخفف نوع داده ذکر می‌شود که یک روش شناخته شده و معروف برای نام گذاری متغیرها است.

چند نمونه نام متغیر دو کلمه‌ای، به روش مجارستانی را ملاحظه می‌کنید :

IntNumber, LngSalary, BlnStatus

در این کتاب از روش کوهان شتری برای نام‌گذاری متغیرها استفاده شده است.

نکته

با توجه به حساسیت زبان C# به حروف کوچک و بزرگ، در نام گذاری متغیرها به این نکته دقت کنید که متغیر a و A مستقل هستند.

۱ــ Came Notat on

۲ــ Hungar an Notat on

۳ــ Char es S mony

در جدول زیر تعدادی نام متغیر و علت مجاز یا غیرمجاز بودن این نام‌ها را می‌بینید.

جدول ۳ــ۴ــ نمونه متغیرهای مجاز و غیر مجاز

نام متغیر	توضیح
1a	غیر مجاز، نام متغیر نباید با عدد شروع شود
a1	مجاز
employee Salary	غیر مجاز، بین کلمات نباید فاصله وجود داشته باشد
First	مجاز
Hello!	غیر مجاز، علامت تعجب نباید در نام وجود داشته باشد
payRate	مجاز
one+two	غیر مجاز، علامت + در یک نام نباید قرار داشته باشد
Conversion	مجاز
counter 1	مجاز
2nd	غیر مجاز، نام نمی‌تواند با عدد شروع شود

در دستورات زیر، چند نمونه از اعلان و مقدار دهی متغیرها را مشاهده می‌کنید :

```
int speed   70 ;  // تعریف متغیر برای نگهداری سرعت خودرو با مقداردهی اولیه
float  a , b , c ;        //  Triangle sides تعریف سه متغیر برای اضلاع مثلث
float  triangleArea ;     //  تعریف یک متغیر برای نگهداری مساحت مثلث
double electricalCharge;  //  متغیری برای نگهداری بار الکتریکی یک جسم
```

۷ــ۴ــ کار با اعداد اعشاری

در فیزیک و شیمی و یا به طور کلی در علوم، با اعداد بسیار کوچک و بسیار بزرگ سروکار داریم. اگر بخواهید عدد اعشاری بسیار کوچک و یا بسیار بزرگی را در یک متغیر ذخیره کنید، می‌توانید آن را به صورت کوتاه با روشی شبیه نماد علمی‌بنویسید. برای اینکه با این روش آشنا شوید ابتدا لازم است روش نماد علمی‌را یادآوری کنیم.

در روش نماد علمی، هر عدد از ۲ بخش تشکیل می‌شود که با علامت ضرب از یکدیگر جدا شده اند. بخش اول یک عدد اعشاری بین ۱ تا ۹ است (فقط یک رقم صحیح دارد) که به آن **مانتیس** می‌گویند و قسمت دوم که به صورت توانی از عدد ۱۰ است که به آن **نما** گفته می‌شود جدول ۴ــ۴).

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();
```

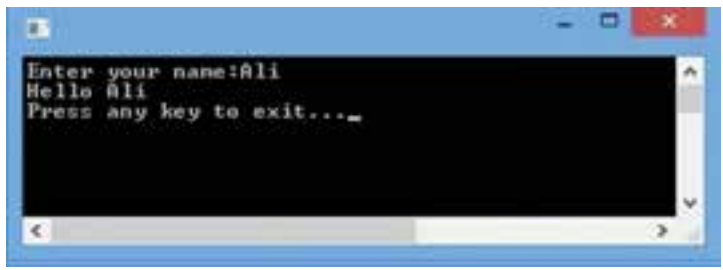
کار در کارگاه ۲

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؟

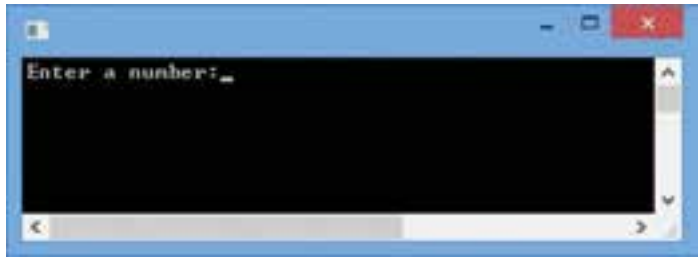
مثال ۵ــ۴ : می‌خواهیم برنامه‌ای بنویسیم که دو عدد دلخواه از کاربر دریافت کند و مجموع آن‌ها را حساب کرده و روی صفحه نمایش، نشان دهد.

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();
    }
}
```

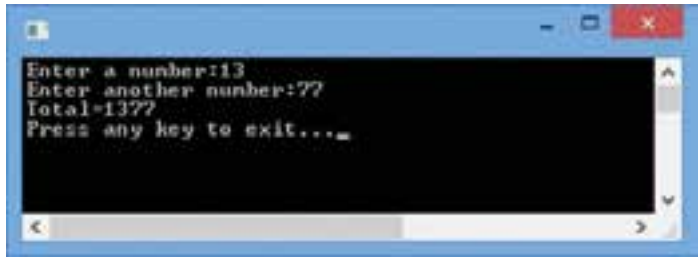
برنامه ۵ــ۴ــ اولین تلاش برای دریافت داده‌های عددی

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؟

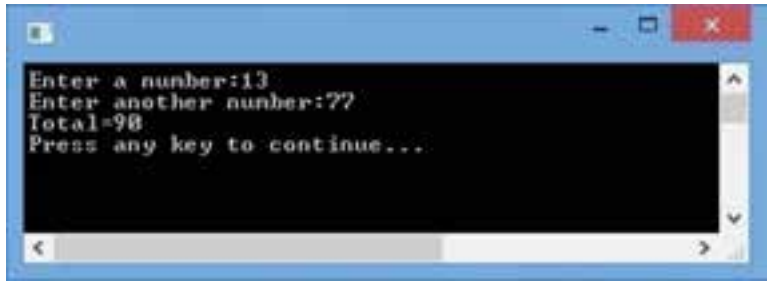
کار در کارگاه ۳

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؟

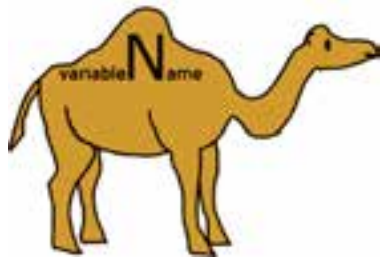
۵ــ تفاوت‌های بین نوع داده float و double را نام ببرید.

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.

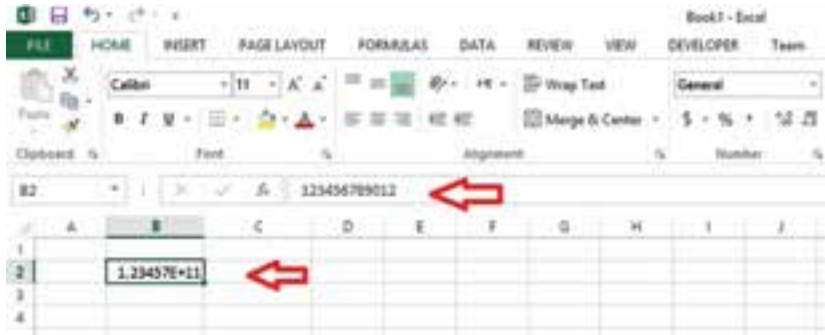
پ) اندازه جرم یک اتم کربن (جرم اتمی) برابر با ۱۶۶۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰ kg

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	