فهرست مطالب

کامپیوتر یک وسیله ی الکترونیکی است که به حل مشکل خاصی می پردازد یا وظیفه ی مشخصی را انجام می دهد. برای مثال، یک دستگاه تناسب اندام به شخص کمک می کند، هیکل خود را رو فرم نگه دارد یا یک دوربین دیجیتال از چیزهای مختلف عکس می گیرد.

وسایل الکترونیکی به منظور حل مسائل عمومی نیز به کار می روند. برای مثال، از رایانه ی شخصی برای انجام کارهای عمومی مثل محاسبات، پردازش کلمه، یا ذخیره سازی بانک های اطلاعاتی استفاده می شود .

برای انجام وظایف مختلف، کامپیوتر باید مقدارهای (value) مشخصی دریافت کند. این کار ممکن است توسط شخصی صورت بگیرد که دستوری را از صفحه کلید تایپ می کند (به عنوان مثال از دستگاه تناسب اندام، تلفن همراه، یا کامپیوتر شخصی). در برخی از موارد، مقادیر موردنظر (از داخل) الکترونیکی و از منابع مختلف به کامپیوتر داده شود .

به منظور مدیریت این ارتباطات، کامپیوتر از صفحه ی تختی به نام motherboard استفاده می کند. خیلی از بخش های کامپیوتر به این تخته ی اصلی متصل هستند و از آن دستور دریافت می کنند. بخش دیگری وجود دارد که وظیفه ی اصلی آن پردازش و انجام محاسبات است که به آن پردازش گر می گویند و به تخته ی اصلی (motherboard) وصل می باشد.

حال، مقادیری که کامپیوتر دریافت می کند باید در قسمتی به نام memیا yیا حافظه ذخیره شود. رایانه از دو نوع حافظه برای این منظور استفاده می کند. حافظه ی موقت و حافظه ی پایدار. از حافظه ی موقت برای ذخیره کردن اطلاعاتی استفاده می شود که حالت موقتی داشته و پس از گذشت زمان مشخصی پاک می شوند. برای مثال، حافظه ی نام برده اطلاعات را هنگامی که رایانه روشن است در خود حفظ می کند و آن را تا زمانی که کامپیوتر روشن است نگه می دارد ولی به محض خاموش شدن رایانه اطلاعات مزبور پاک می شوند.

حافظه ای که اطلاعات و مقادیر به صورت موقت در آن ذخیره می شود random access memیا y یا RAMنامیده می شود.

فرض کنید، حافظه ی کامپیوتر یک سینی کیک است متشکل از چند بخش، که هریک حامل چیزی است.

توجه داشته باشید که حافظه بزرگ تر است یک کیک است و از میلیون ها جای خالی تشکیل شده.برنامه نویس مدام به compiler دستور می دهد که مقادیر را به صورت موقتی در RAM ذخیره کند. اگرچه اندازه ی حافظه ی موقت چندان بزرگ نیست، باید به خاطر داشته باشید جای زیادی برای ذخیره ی اطلاعات لازم دارد. در واقع، این تنها برنامه ی شما نیست که از RAM استفاده می کند. برای مثال، هنگامی که رایانه ی خود را راه اندازی می کنید، سیستم عامل (windows) و دیگر برنامه ها ی آن RAM را اشغال می کنند.

هنگامی که برنامه ای را اجرا می کنید، compilerبخشی از RAM را به آن برنامه اختصاص می دهد.

به این خاطر که برنامه های زیادی از RAM استفاده می کنند، به منظور ذخیره سازی مقادیر در آن، باید اطلاعاتی در اختیار آن قرار دهید .باید مقدار حافظه ای را که به آن نیاز دارید مشخص کرده و اسم معینی برای آن قسمت خاص از حافظه که value ها در آن ذخیره می شود انتخاب کنید. به ترکیبی از این اطلاعات متغییر (variable) گفته می شود.

معرفی متغییرها

- .1برنامه ی Microsoft Visual C# 2010 Express یا Microsoft Visual Studio 2010را اجرا کنید .

- .2به منظور ایجاد برنامه ی کاربردی (app) جدید، در Start Page روی گزینه ی New Project کلیک کنید .
- .3در فهرست میانی، گزینه ی Empty Project را انتخاب کنید .
- .4اسم موردنظر را به gdcs2(ge)یا ge town cleaning servicesتغییر دهید .
- .5روی ok کلیک کنید .
- .6به نظور ایجاد فایل ویژه ی کد مورد نظر، به main menu مراجعه کرده و project را انتخاب کنید .
- .7از لیست میانی code file را انتخاب کنید .
- .8اسم نام برده را به cleaning یا der تغییر دهید .
- .9روی گزینه ی Add کلیک کنید .
- .10در داکیومنت خالی دستورات زیر را تایپ کنید .

?

```
1         class Order
2{
3     static void Main(string[] args)
4     {
5         System.Console.WriteLine("Georgetown Dry
6Cleaning Services");
7         System.Console.ReadKey();
8     }
9}
```

اسم ها

متغییرها باید اسم مشخصی داشته باشند و برای ایجاد اسم برای آن ها باید از قوانین خاصی پیروی کرد. کلماتی وجود دارند که از آن ها نباید تحت هیچ شرایطی به عنوان اسم متغییر استفاده کرد زیرا خود برنامه کلیدواژهای مزبور را به کار می برد. این کلیدواژه ها در زیر فهرست شده.

abstract	continue	finally	interface	out ‹generic›	short	typeof
as	decimal	fixed	internal	out ‹methods›	sizeof	uint
base	default	float	is	override	stackalloc	ulong
bool	delegate	for	lock	params	static	unchecked
break	do	foreach	long	private	string	unsafe
byte	double	goto	namespace	protected	struct	ushort
case	else	if	new ‹generic›	public	switch	using
catch	enum	implicit	new ‹LINQ›	readonly	this	virtual
char	event	in ‹foreach›	new ‹variable›	ref	throw	void
checked	explicit	in ‹generic›	null	return	true	volatile
class	extern	int	object	sbyte	try	while
const	false		operator	sealed		

واژه های دیگری هستند که با وجود این که جز کلیدواژه های برنامه ی C# نیستند، استفاده از آن ها به شما توصیه نمی شود. به این خاطر که به کار بردن آن ها ممکن است منجر به بروز اختلال (conflict) در کد شود .از واژه های داده شده به عنوان کلیدواژه های متنی (contextual keyw یا(ds یاد می شود.

add	global	let	partial ‹type›	set	where ‹generic›
dynamic	group	orderby	remove	value	where ‹query›
from	into	partial ‹method›	select	var	yield
from	from				

همان طور که پیش تر ذکر شد، اسم گذاری برای هر چیزی در برنامه نویسی قوانین خاص خود را دارد.قوانین استانداردی هست که توسط C# تعریف می شوند ولی شما می توانید قوانینی را بر مبنای سلیقه ی خود نیز به وجود بیاورید. قوانینی که برای تعیین اسم متغییر باید پیروی کنید به شرح زیر می باشد.

اسم می تواند از تنها یک حرف تشکیل شود(A، B، C، D، E، F، G، H، I، J، K، L، M، N، O، P، Q، R، S، T، U، V، W، X، Y، Z، a، b، c، d، e، f، g، h، i، j، k، l، m، n، o، p، q، r، s، t، u، v، w، x، y، یا z). این حروف متعلق به الفبای انگلیسی آمریکایی می باشد، ولی زبان C# انگلیسی بین المللی را نیز می پذیرد.

درC# ، اسم ممکن است از تنها یک علامت (_) ساخته شود. با این وجود، توصیه می شود از این کار خودداری کنید زیرا خواندن کدی که از تنها یک _ تشکیل شده آسان نیست.

از به کار بردن این علامت ها در اسم خودداری کنید| .، !، "، /، $، %، ?، &، *،)، (، +، #، \، @، >،<،]، [، ;، }، {

در صورتی که اسمی متشکل از بیش ازیک کاراکتر باشد، باید با حرف یا _ شروع شود.

پس از آغاز کاراکتر با _ یا حرف، اسم می تواند از ترکیبی از حروف، ارقام (۰، ۱، ۲، ۳، ۴، ۵، ۶، ۷، ۸، یا ۹)، و / یا _ تشکیل شود.

در اسم نمی توان از فاصله (space) استفاده کرد.

علاوه بر قوانین بالا، برنامه نویس می تواند قوانین خود را تعریف کند. ولی در تعریف همین قوانین باید محدودیت های ذکر شده در نظر گرفته شود.

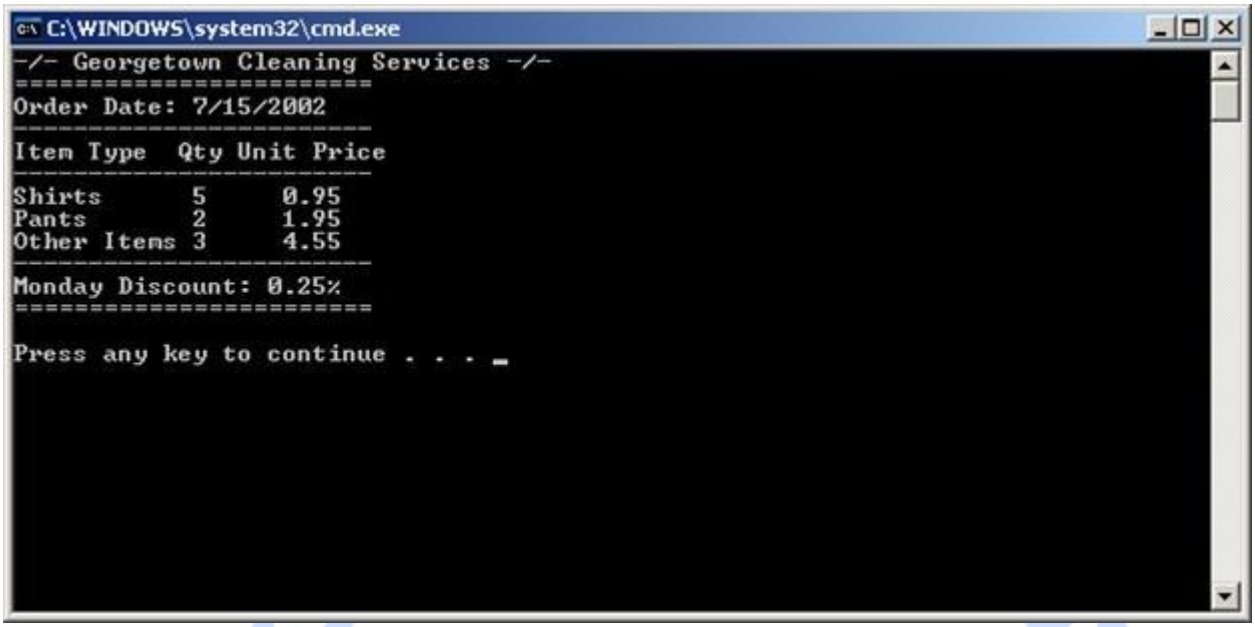
اگر اسمی در تنها یک کلمه خلاصه شود، اکثر برنامه نویسان از حروف کوچک استفاده می کنند.

زمانی که اسم متغییر متشکل از چند کلمه است، بیشتر برنامه نویسان از camel notation استفاده می کنند که در آن اولین کلمه با حروف کوچک نوشته می شود و اولین حرف کلمات بعدی همگی با حروف بزرگ نوشته می شوند .

زبان C# به کوچک بزرگی حروف حساس است، یعنی کلماتcase ، Caseو CASE با هم کامل متفاوت اند. برای مثال، mainهمیشه Main نوشته می شود.

مقدارها و متغییرها در کنسول

برنامه های زبان C# نتایج خود را در پنجره ی DOS به نمایش می گذارند. به مثال زیر توجه کنید.



برای نشان دادن مقدار معینی در پنجره ی فوق، می توانید آن را داخل پرانتز System.Console.Write() یا System.Console.WriteLine()قرار دهید. مانند دو نمونه ی زیر

```
1
2         class Exercise
3{
4     static void Main()
5     {
6         System.Console.WriteLine(248);
7         System.Console.Write(1);
8     }
9}
```

اگر داخل پرانتز System.Console.WriteLine() را خالی بگذارید، خطی تهی نمایش داده می شود.

نمایش های عددی

مقدمه

برنامه ی کامپیوتر عبارتند از مجموعه ای دستور که برای رایانه انجام کار مشخص، زمان اجرای آن و نحوه ی پیاده سازی آن را معیین می کند. این برنامه نویس است که دستورها را می نویسد. همان طور که پیش تر نیز بیان شد، می توان دستورات مزبور را در برنامه ی text edit یا، البته بر مبنای قوانین استاندارد زبان C# ولی با زبان شناس مانند انگلیسی نوشت. برای مثال، می توان دستوری نوشت که از کامپیوتر درخواست می کند عددی را در حافظه ذخیره کند. تمام دستوراتی که می نویسید باید به کامپیوتر منتقل شود.

مستحضر هستید که اشخاص دستورهای مختلفی می نویسند. همچنین، C#تنها زبان برنامه نویسی نیست، و برنامه نویسان سرتاسر دنیا از زبان های مختلف برای دادن دستور به رایانه استفاده می کنند. کامپیوتر برای درک تمام این زبان ها و تفسیر تمام دستورهایی که از این برنامه ها ارسال می شود، از زبان مختص به خود استفاده می کند که تمام برنامه ها باید آن را بشناسند و از آن پیروی کنند. زبان های داده شده آن جور که باید از مسئله ی فوق پیروی نمی کنند. برای مثال، C#که از زبان انگلیسی استفاده می کند، دستورات خود را به برنامه ی میانجی / واسط ای به نام Assembler تحویل می دهد assembler.دستورات را به نسخه ی فشرده تر تبدیل می کند که با وجود استفاده از لغات جدید، آن هم به زبان انگلیسی می باشد. کامپیوتر دوباره نسخه ی فشرده را به ورژنی ساده تر تبدیل می کند و در اختیار رایانه قرار می دهد.

زبان ساده شده ای که کامپیوتر استفاده می کند، متشکل از ترکیبی از ۱ ها و ۰ ها می باشد، یعنی assemblerباید دستورات c# را به ۰ و ۱ تبدیل کند تا کامپیوتر بتواند آن را تفسیر کرده و اجرا کند.

سیستم های عددی

با سیستم هایی که از ده رقم ۰، ۱، ۲، ۳، ۴، ۵، ۶، ۷، ۸ و 9استفاده می کنند آشنایی دارید. ترکیبی از این ده عدد می تواند هر رقمی که شما دوست دارید بسازد. به این خاطر که زبان کامپیوتر از ۱۰ رقم استفاده می کند به آن سیستم decimal یا دهگانی می گویند.

همان طور که پیش تر ذکر شد، زبان کامپیوتر از ترکیب دو مقدار ۰ و۱ استفاده می کند. به چنین سیستمی، سیستم دوتایی یا binary می گویند. ۱، ۱۰، ۱۰۰۱یا ۱۰۱۰۱۱۰۱۱۱ نمونه هایی از این سیستم می باشند. هر مقداری که به کامپیوتر داده می شود باید ترکیبی از ۰ و ۱ ها تبدیل شود.

همان طور که تصور می کنید نمایش دادن یک رقم بزرگ کار بسیار دشواری است. یک روش برای این منظور، از ۱۰عدد سیستم دهگانی و ۶ حرف از الفبای انگلیسی استفاده می کندA ،. B، C، E، D، و F یاa ، b، c، d، e و .f به این معنا که سیستم فوق از ۱۶ کاراکتر برای ایجاد یک رقم استفاده می کند .به همین دلیل است که به آن سیستم hexadecimal یا مبنای ۱۶می گویند. برای این که فرایند تشخیص یک رقم مبنای ۱۶ از کلمه سهل شود، رقم داده شده باید با ۰ xآغاز شود.نمونه های آن به این صورت است. x۲۰، 0xED8، 0x962AAED3.

علامت دار و بدون علامت

یک از ویژه گی های رقم این است که مشخص می کند آیا از ۰ کوچکتر است، با آن برابر است یا از آن بزرگتر است. عددی که کوچکتر از صفر است منفی محسوب می شود. چنین رقمی به دنبال علامت – می آید. چنانچه، عددی از صفر بزرگتر بود مثبت تلقی می گردد. چنین رقمی به دنبال علامت + می آید .به ارقامی

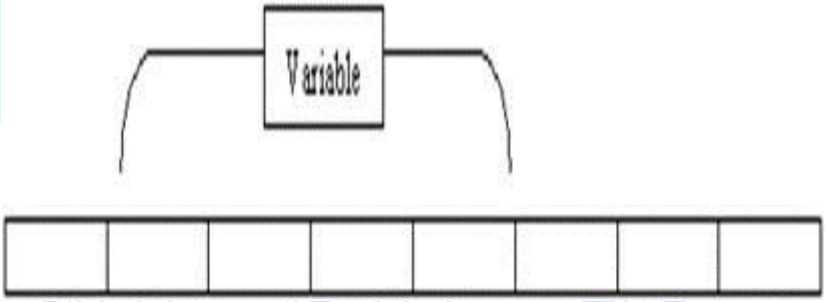
که یکی از این دو علامت را به یدک می کشند signed یا علامت دار می گویند. هر عددی هم که هیچ یک از این دو علامت را نداشته باشد unsigned یا مثبت اتلاق می گردد. برای مثال، عدد ۰ بدون علامت یا unsignedاست.

تعریف متغییرها

Representing Mem یا Bits

همان طور که پیش تر ذکر شد، برای ذخیره سازی مقدار (value) معینی در حافظه ابتدا باید مقدار حافظه ی مورد نیاز را مشخص کنید، سپس باید برای آن اسم انتخاب کنید. مقدار حافظه ی مورد نیاز را data type) نوع داده) می گویند. مقدارهای متفاوت، مقدار حافظه ی متفاوت می طلبد. این را هم گفتیم که رایانه مقدار را ۰ و ۱ می بیند. آن قسمت حافظه که تنها یک ۰ یا ۱ در خود جای می دهد bit خوانده می شود.

تصور کنید که bit یک شئ تهی است که می تواند خالی یا پر باشد.



هنگامی که آن شی خالی است، bitمقدار ۰ را نشان می دهد و زمانی که پر است، مقدار ۱.

تعریف متغییر

با به کاربردن ترکیبی از bit ها، به compiler پیغام می دهید که به متغییر نیاز دارید. از چنین فرایندی به عنوان (declaring a variable) تعریف متغییر یاد می شود. همان طور که قبل توضیح داده شد، مقدار حافظه ای که به آن نیاز دارید data type گفته می شود. بنابراین، هنگام تعریف متغییرباید نوع داده ی (data type)دلخواه را مشخص کنید. در#C ، نوع داده (data type) با یک کلیدواژه نمایش داده می شود.

برای تعریف متغییر دو راه پیش رو دارید.

۰اگر نوع مقداری که مایلید در حافظه ذخیره شود را می دانید، از فرمول زیر پیروی کنید .

?

```
DataTypeVariableName;
```

·اگر نمی خواهید نوع مقدار را مشخص کنید، گزینه های دیگری برای شما فراهم است.

می توانید از کلید واژه ی var استفاده کنید.

می توانید کلید واژه ی object را به کار ببرید .

می توان از کلیدواژه ی dynamic نیز استفاده کرد.

مقدار دهی اولیه ی متغییر

همان طور که از اسم آن پیدا است، مقدار دهی اولیه عبارتند از ذخیره سازی مقدار اولیه در جای (مشخص آن. (بیش تعریف شده به منظور اختصاص دادن متغییر.

نوع داده، به دنبال آن اسم متغییر، علامت =، مقدار دلخواه ولی مناسب را تایپ کنید.

?

```
                class Exercise
{
    static void Main()
    {
        DataTypeVariableName = DesiredValue;
    }
      }
```

با این روش، شما می توانید متغییر را در ابتدای کار با مشخص کردن اسم و نوع داده تعریف کنید و در انتهای آن علامت نقطه ویرگول قرار دهید. سپس، در خطی دیگر، اسم متغییر را تایپ کرده و به دنبال آن علامت =، و مقدار مورد نظر را وارد کنید .

?

```
    class Exercise
{
    static void Main()
    {
        DataTypeVariableName;
        VariableName = DesiredValue;
    }
}
```

ابتدا کلیدواژه ی var را تایپ کرده و به دنبال آن، اسم متغییر، علامت = و مقدار معین را درج کنید .

?

```
1
2                class Exercise
3{
4     static void Main()
5     {
6          varVariableName = DesiredValue;
7     }
8          }
```

این بار باید تمام مراحل را یکجا انجام دهید. در این مورد، نمی توان ابتدا اسم متغییر را با قرار دادن var و نقطه ویرگول (;) تعریف کرده و بعد (در خطی دیگر) مقدار دهی اولیه متغییر را انجام دهید. با این کار، پیغام خطا دریافت می کنید.

?

```
1
2                    class Exercise
3    {
4         static void Main()
5         {
6              varVariableName;
7
8              VariableName = DesiredValue;// Error
9         }
10}
```

دو گزینه ی پیش روی دیگر نیز، استفاده از کلیدواژه های object یا dynamic می باشد.

پس از تعریف و مقدار دهی اولیه ی متغییر، compilerمقدار آن متغییر را حافظه ای که برای آن اختصاص یافته ذخیره می شود. حال، می توان به آن مقدار دسترسی پیدا کرد و در صورت نیاز آن را تغییر داد.

مقدمه ای بر مقدارهای برنامه قسمت دوم

مقدار تهی (null value)

می توان با اضافه کردن علامت سوال (؟) به نوع داده (data type) متغیر موردنظر، متغیر را طوری تعریف کنید که مقدار تهی (null value) داشته باشد. برای این منظور زبان C# کلیدواژه ی null را در اختیار شما قرار می دهد.

می توان هنگام تعریف متغیر، مقدار null را به آن اختصاص داد.

?

```
1
2         class Exercise
3{
4static void Main()
5             {
6         DataType?VariableName= null;
7         // You can use the variable
8             }
9         }
```

همچنین می توان این مقدار را پس از تعریف متغیر اختصاص داد.

?

```
1
2          class Exercise
3{
4static void Main()
5             {
6         DataType?VariableName;
7         VariableName= null;
8         // You can use the variable
9             }
10        }
```

آنچه اهمیت دارد آن است که مقدار پیش از به کاربردن متغیر به آن اختصاص داده شود.

Byte

ترکیبی از چهار bit

اگرچه یک bit قابل دسترس و استفاده است، باید به خاطر داشت که نمی توان در آن مقدار ذخیره کرد به این معنا که نمی توان از compiler خواست که مقدار ۱ را در یک bit ذخیره کند. این امر به این خاطر است که، حتی کوچکترین مقدار C# نیز به بیش از یک bit برای ذخیره شدن نیاز دارد. ترکیبی کوچکتر از چهار bit وجود ندارد (در برخی زبان ها یا پیاده سازی زبان assembly به ترکیب چهار bit ، nibble گفته می شود.)

با ایجاد ترکیبات چهارتایی از پر (۱) و تهی (۰)، همگی ۱۶ ترکیب به دست می آید.

در این ترکیبات چهارتایی، bitها 0، 1، 2و ۳ شمرده می شوند .bit ای که در راست ترین کناره یا موقعیت قرار دارد) ۰(، bitرده پایین خوانده می شود.(LOBIT)

آخرین bit یا bit ای که در چپ ترین موقعیت قرار دارد) ۳(، bitرده بالا خوانده می شود.(HIBIT)

اگر بخواهیم ترکیبات چهار bit ای را با سیستم دودویی (binary)نمایش دهیم به این نتایج دست می یابیم ۰۰۰۰، ۰۰۰۱، ۰۰۱۰، ۰۰۱۱، 0100، 0101، 0110، 0111، 1000، 1001، 1010، 1011، 1100، 1101، 1110، 1111، که همگی ۱۶ ترکیب به دست می آید. با فرمت دهدهی (decimal) ترکیبات بالا این نتایج را به دست می دهد : ۰، ۱، ۲، ۳، ۴، ۵، ۶، ۷، ۸، ۹، ۱۰، ۱۱، ۱۲، ۱۳، 14، و ۱۵. حال، با فرمت hexadecimal)مبنای ۱۶(

Decimal	Binary	Hexadecimal
0	0000	0x0
1	0001	0x1
2	0010	0x2
3	0011	0x3
4	0100	0x4
5	0101	0x5
6	0110	0x6
7	0111	0x7
8	1000	0x8
9	1001	0x9
10	1010	0xA
11	1011	0xB
12	1100	0xC
13	1101	0xD
14	1110	0xE
15	1111	0xF

جدول تبدیل عددی

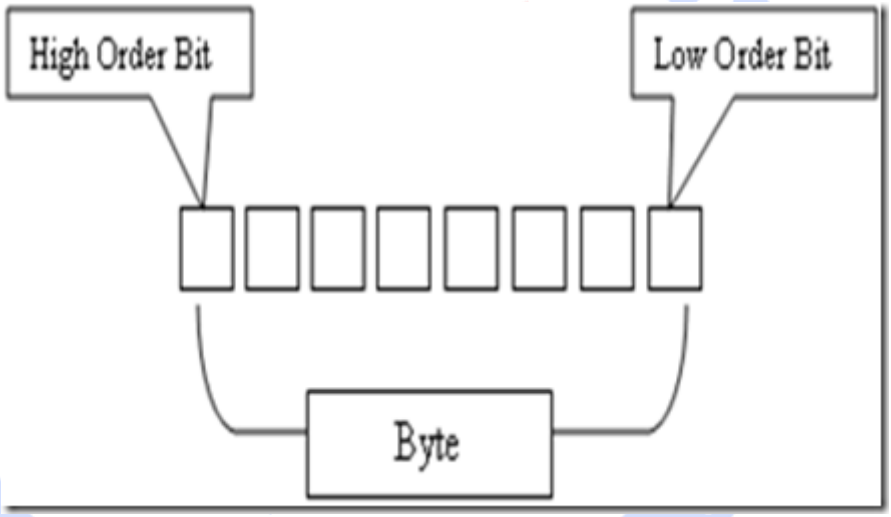
حداقل و حداکثر مقدارها در ترکیب چهار bit ای به این صورت است.

	Decimal	Hexadecimal	Binary
Minimum	0	0x0	0000
Maximum	15	0xf	1111

البته، باید در نظر داشته باشید که با ترکیبات چهارتایی هیچ کاری نمی توان کرد. زیرا ترکیبات بیان شده بسیار کوچک هستند و گنجایش ذخیره هیچ چیز را ندارد. به دو دلیل درباره ی آن بحث کردیم : اول اینکه، باید بدانید که عدد ۱۶ از کجا سر رشته می گیرد. دوم اینکه، پیش زمینه ی معرفی byte را فراهم آورد.

ترکیب ۸ bit

یک byte ترکیبی از ۸ bit است که کنار هم قرار گرفته. اولین bit ای که در راست ترین قسمت قرار گرفته) ۰ و آخرین 7 bit می باشد.



0کم اهمیت ترین bit است که به آن LOBIT نیز می گویند. ۷ مهم ترین bit محسوب می شود و نام دیگر آن HIBITاست.

ترکیبات ۸ تایی با سیستمbinary: 00000000 ، 00000001، 00000010، 00000011، 00000100، 00000101، 00000110، 00000111، 00001000، 00001001، 00001010، 00001011، 00001100، 00001101، 00001110، 00001111تا .11111111نمایش دادن یک عدد بزرگ با فرمت دودویی، خواندن آن را سخت می کند. روش مناسب تر، دسته بندی آن ها در گروه های چهارتایی است به این صورت 01001011 = 0100 1011

به منظور ارزیابی تعداد ترکیبات در فرمت دهدهی(decimal) ، عدد ۲) که نشانگر decimal است(، را به توان bit موردنظر (۰، ۱، ۲، ۳، ۴، 5، 6، (7برده، سپس عدد را اضافه می کنیم به این صورت

?

```
27 + 26 + 25 + 24 + 23 + 22 + 21 + 20
= 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1
                                = 255

```

بنابراین، ۲۵۵ ترکیب احتمالی ۸ bit ای داریم. ترکیبات بالا را می توان با فرمت hexadecimal نیز حساب کرد : ۰x1، ۰x2،....0xA، 0xA1،....0xC8،.... up to 0xFFFF

محاسبات در سه سیستم عددی مختلف

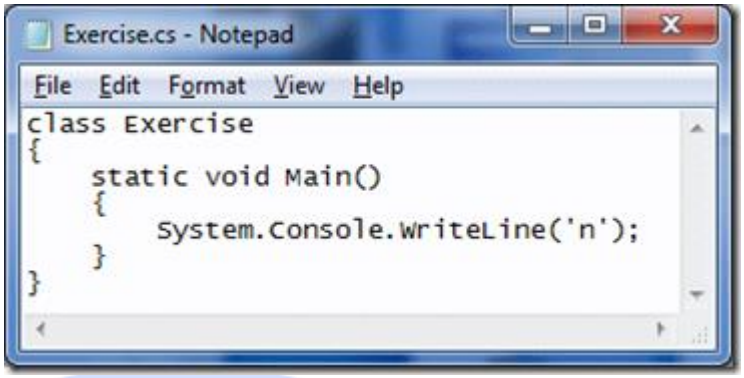
	Decimal	Hexadecimal	Binary
Minimum	0	0x0	0000
Maximum	255	0xff	1111 1111

حداقل مقدار حافظه اختصاص داده شده (توسط intel computer)به منظور ذخیره سازی byte می باشد. همان طور که مستحضر هستید، یک byteمتشکل از ۸ bit متوالی و کنارهم قرار گرفته است. مقدار حافظه ای که به اندازه ی یک byte در اختیار شما قرار داده می شود، تنها گنجایش ذخیره کردن یک نشانه (که بر روی صفحه کلید درج شده است) را دارد. این نشانه ها، که به آن ها کاراکتر هم گفته می شود، توسط ASCII، کد استاندارد آمریکایی به منظور تبادل اطلاعات تعبیه شده اند. اما باید در نظر داشت که ASCII از تنها ۱۲۸ عدد) decimal بر مبنای فرمت ۷ bit ای (استفاده می کند (از ۰ تا ۱۲۷) .

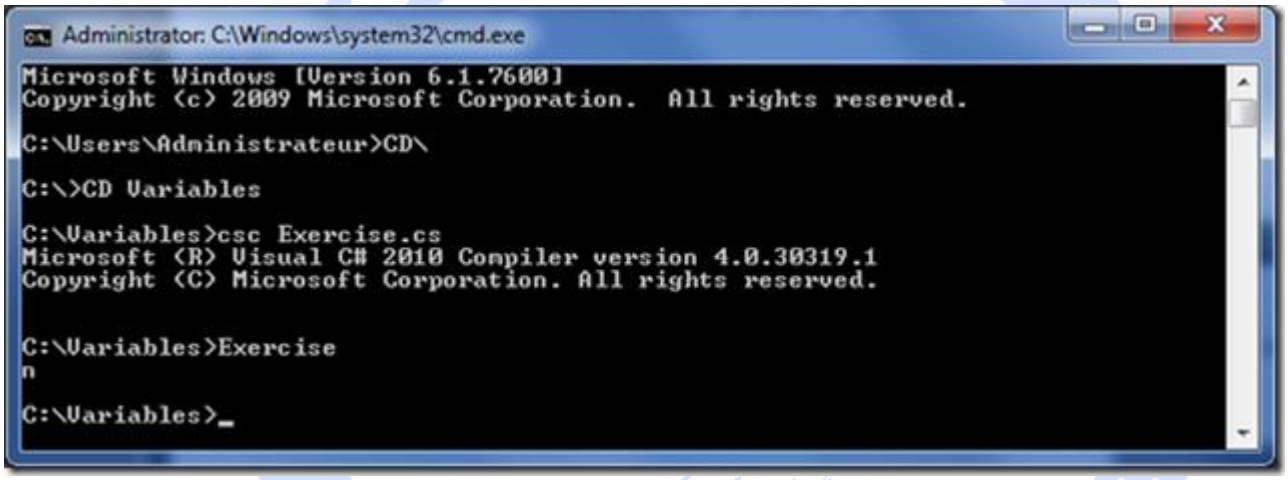
کلیه ی کاراکترهایی که در صفحه کلید مشاهده می کنید، به عنوان یک مقدار عددی نمایش داده می شود، اما هریک از این نشانه ها چه عدد، چه حرف و چه علامت همگی کاراکتر محسوب می شوند .برای نمایش دادن هر کاراکتری در صفحه ی نمایش، می توان آن را با ()Write یا()WriteLine ، ارسال کرد (pass) و کاراکتر مورد نظر را در علامت (' ') قرار داد.

```
        class Exercise
{
            static void Main()
            {
        System.Console.WriteLine('n');
            }
        }


```



فایل موردنظر با نام Exercise.cs در فولدر Variables در درایو C:\ ذخیره گشت. پس از ترجمه (compile)و اجرا شدن، حرف n در صفحه نمایش داده می شود.



کاراکترها

در الفبای زبان انگلیسی، حرف به یکی از نشانه های زیر گفته می شود: a ، b، c، d، e، f، g، h، i، j، k، l، m، n، o، p، q، r، s، t، u، v، w، x، y، z، A، B، C، D، E، F، G، H، I، J، K، L، M، N، O، P، Q، R، S، T، U، V، W، X، Y و Z. جدا از این کاراکترهای خوانا، به علامت های ذیل نیز رقم (digit) می گویند : ۰، ۱، ۲، ۳، ۴، ۵، ۶، ۷، ۸، ۹. علامت های دیگری وجود دارند که به آن ها نیزکاراکتر می گویند ` : > ". ،~ ! @ # $ % ^ & * ( ) - _ = + [{] } \ | ; : ' < ?. /

زبان C# هر چیزی را که بتوان به صورت نشانه ای به نمایش گذاشت را یک کاراکتر به حساب می آورد. به منظور تعریف متغیری که مقدارش یک کاراکتر می باشد، می توان از کلیدواژه ی var استفاده کرد و متغیر را با کاراکتری درون علامت ' ' مقدار دهی اولیه (initialize) کرد. به مثال زیر توجه کنید.

?

```
1 class Exercise
2 {
      static void Main()
```

```
3               {
4                   var gender = 'F';
5                     System.Console.Write("Student Gender: ");
6              System.Console.WriteLine(gender);
7               }
8           }
9
10
```

نتیجه ی زیر حاصل می گردد.

همچنین، می توان از کلیدواژه ی char استفاده کرد. توجه خود را به مثال زیر جلب کنید.

?

```
1
2           class Exercise
3{
4               static void Main()
5               {
6           char gender = 'M';
7           System.Console.Write("Student Gender: ");
8           System.Console.WriteLine(gender);
9           System.Console.ReadKey();
10              }
11          }
```


توجه :

تمام زبان هایی که بر مبنای C فعالیت می کنند نوع داده ی char را پشتیبانی می کنند.

نتیجه ی زیر به دست می آید.

?

```
1         Student Gender: M
2           Escape Sequence
3
```

Escape sequenceکاراکتری ویژه است که در صفحه نمایش قابل رویت نمی باشد. برای نمونه می توان از این کاراکتر برای شروع نوشتن دستور در خط بعدی استفاده کرد (به برنامه فهماند که این خط به اتمام رسیده و خط دیگری باید شروع شود). کاراکتر مزبور با این علامت /نشان داده می شود که به دنبال آن ممکن است یک کاراکتر یا نشانه ی دیگر قرار گیرد. برای مثال، escape sequenceای که در خط بعدی قرار می گیرد به این صورت نمایش داده می شود \n :

کاراکتر escape sequence ممکن است داخل ' ' قرار داده شود .'\n' : هم چنین این کاراکتر را می توان داخل " ". قرار داد ."\n" :

توجه :

تمام زبان هایی که مبنای آن C هست، این کاراکتر ها را پشتیبانی می کنند.

Escape Sequence	Name	Description
\a	Bell (alert)	صدایی از کامپیوتر پخش می کند
\b	Backspace	نشان گر موس را به عقب می برد
\t	Horizontal Tab	نشان گر موس را به صورت افقی به حرکت در می آورد
\n	New line	نشان گر موس را به آغاز خط بعدی می برد
\v	Vertical Tab	جدول بندی عمودی ایجاد می کند
\f	Form feed	
\r	Carriage return	فرایند بازگشت به ابتدای خط را فراهم می سازد
\"	Double Quote	(") علامت نقل و قول را نمایش می دهد
\'	Apostrophe	(') علامت آپستروف را نمایش می دهد
\?	Question mark	علامت سوال را نشان می دهد
\\	Backslash	را نشان می دهد (\) علامت
`	Null	کاراکتر تهی را نمایش می دهد

برای استفاده ازEscape sequence ، همچنین می توان یک متغیر charمعرفی کرده، آن را با کاراکتر escape sequenceدلخواه مقداردهی اولیه کرد، سپس داخل ' 'قرار داد.

مقدمه ای بر انواع داده در سی شارپ قسمت سوم

نوع داده ی Byte

یک byte عددی بی علامت (unsigned number) است که مقدار آن از ۰ تا ۲۵۵ متغیر می باشد، به این خاطر هیچ چیز در یک byte قابلیت ذخیره شدن ندارد. برای معرفی متغیری که دربردارنده ی یک عدد طبیعی کوچک است، می توان از کلیدواژه ی byte استفاده کرد مانند مثال زیر

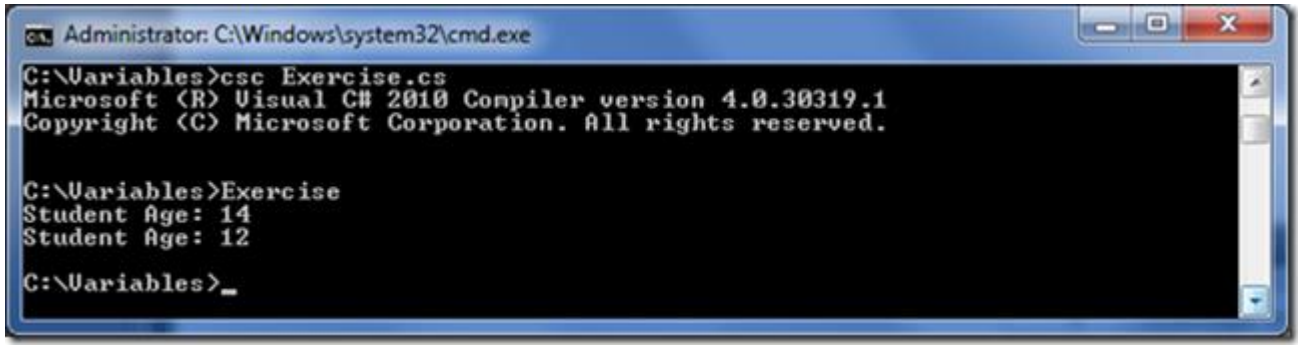
byte Age

می توان متغیر byte را در حین معرفی متغیر یا پس از انجام فرایند (معرفی) مقدار دهی اولیه کرد. مثال زیر نوع داده ی byte را به کاربرده .

?

```
1
2         class Exercise
3{
4             static void Main()
5             {
6         byte age = 14;
7         System.Console.Write("Student Age: ");
8         System.Console.WriteLine(age);
9
10        age = 12;
11        System.Console.Write("Student Age: ");
12        System.Console.WriteLine(age);
13            }
14        }
```

که محصول زیر را به دست می دهد .



به هیچ وجه از مقداری بالای ۲۵۵ برای متغیر byte استفاده نکنید، زیرا در آن صورت error دریافت می کنید. همچنین، می توان از کلیدواژه ی var برای معرفی متغیر استفاده کرد، سپس آن را با عددی کوچک مقداردهی اولیه کرد. به این مثال توجه کنید.

?

```
1
2       class Exercise
3{
4           static void Main()
5           {
6               var age = 14;
7               System.Console.Write("Student Age: ");
8               System.Console.WriteLine(age);
9               age = 12;
10              System.Console.Write("Student Age: ");
11              System.Console.WriteLine(age);
12      System.Console.ReadKey();
13          }
14      }
```

به جای عدد دهدهی(decimal number) ، می توان متغیر انتگرال را با مقدار hexadecimal (مبنای۱۶) مقداردهی اولیه کرد. ابتدا، اطمینان کسب کنید که معادل دهدهی (decimal equivalent) از مقدار ۲۵۵ پایین تر است. به مثال زیر توجه کنید.

?

```
1
2       class Exercise
3{
4           static void Main()
5           {
6               var number = 0xFE;
7               System.Console.Write("Number: ");
8               System.Console.WriteLine(number);
9       System.Console.ReadKey();
10          }
11      }
```

نتیجه ی زیر حاصل می گردد.

```
1               Number: 254
2Press any key to continue…
3
```

استفاده کردن از Byte

فایل Program.cs را به این صورت تغییر دهید.

```
 1
 2        class Order
 3{
 4            static void Main()
 5            {
 6                byte? shirts = null;
 7                byte? pants  = null;
 8                shirts = 4;
 9                pants  = 1;
10                System.Console.WriteLine("-/- Georgetown Cleaning
Services -/-");
11                System.Console.WriteLine("========================");
12                System.Console.WriteLine("Item Type  Qty");
13                System.Console.WriteLine("------------------------");
14                System.Console.Write("Shirts      ");
15                System.Console.WriteLine(shirts);
16                System.Console.Write("Pants       ");
17                System.Console.WriteLine(pants);
18                System.Console.WriteLine("========================");
19                System.Console.ReadKey();
20            }
21        }
```

به منظور اجرای برنامه به main menu مراجعه کرده، سپس گزینه ی Debug -> Start Debugging را انتخاب کنید. محصول زیر به دست می آید.

```
1
2        -/- Georgetown Cleaning Services -/-
3========================
4Item Type  Qty
5------------------------
6Shirts      4
7Pants       1
8========================
```

حال، برای بستن پنجره ی DOS دکمه ی Enter را فشار دهید.

Byteعلامت دار

یک عددِbyte ، زمانی علامت دار معرفی می شود که بتواند مقداری مثبت یا منفی از ۱۲۸- تا ۱۲۷ را در خود نگه دارد، این مقدار در یک byteذخیره می شود. به منظور معرفی متغیری برای مقدار گفته شده، از کلیدواژه ی sbyte استفاده کنید. نمونه ی آن را در زیر مشاهده می کنید.

```
class Exercise
{
    static void Main()
    {
sbyte roomTemperature = -88;
the room temperature was ،       System.Console.Write("When we entered
");
System.Console.WriteLine(roomTemperature);
    }
}
```

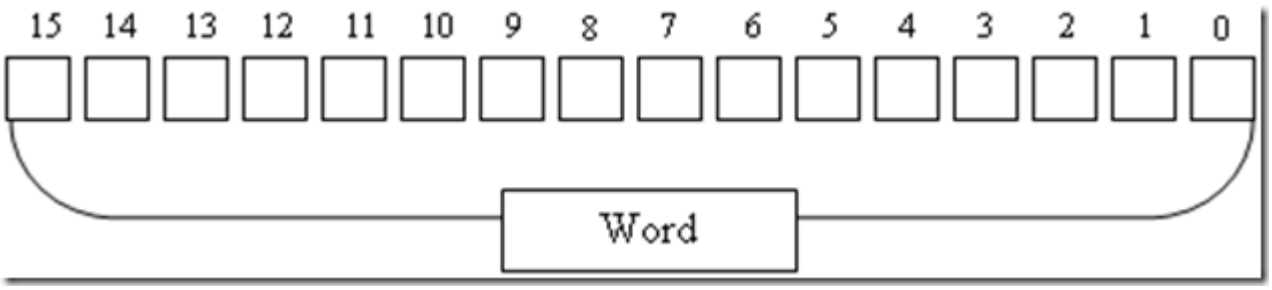
که نتیجه ی زیر از آن به دست می آید.

```
the room temperature was -88،When we entered
```

واژه (Word)

مقدمه

یک واژه متشکل از ۱۶ bit متوالی است .ها از راست به چپ شمرده می شوند (از ۰).



راست ترین bit یک واژه ۰ می باشد که به آن کم اهمیت ترین bitیا bit سطح پایین و یا LOBIT می گویند. چپ ترینbit ، 15، مهم ترین bit یا bit سطح بالا و یا HIBIT خوانده می شود .bit های دیگر بسته به موقعیتی که در آن قرار گرفته اند تعریف می شوند : ۱، ۲، ۳ و غیره.

نظر به این که یک واژه از دو byte تشکیل شده، گروه ۸ bit سمت راست همان LOBYTE خوانده می شود و دیگر گروه ۸ bit ای که در سمت چپ قرار می گیرد HIBYTE نام دارد.

نمایش یک واژه با فرمت (قالب) دودویی (binary) به این صورت است : ۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰. برای خواناتر کردن آن، می توان bit ها را در گروه های چهارتایی قرار داد به این صورت : ۰۰۰۰ ۰۰۰۰ ۰۰۰۰ ۰۰۰۰. بنابراین، کمترین مقدار دودویی که واژه قادر به نمایش دادن آن است به این شکل خواهد بود : ۰۰۰۰ ۰۰۰۰ ۰۰۰۰ ۰۰۰۰. کمترین مقدار دهدهی (decimal) یک واژه معادل ۰ می باشد. کمترین مقدار مبنای ۱۶ (hexadecimal) که می توان در یک واژه ذخیره کرد برابر با ۰ x0000000000000000می باشد که البته به این اشکال نیز نمایش داده می شود : ۰ x00000000یا ۰ x0000و یا ۰ .x0تمام این ارقام یک مقدار را به دست می دهند و آن ۰ x0است.

حداکثر مقدار دودویی (binary) که یک واژه می تواند نشان دهد معادل 1111 1111 1111 1111 : می باشد. برای به دست آوردن حداکثر مقدار دهدهی (decimal) یک واژه ، می توانید از فرمول پایه ۲ استفاده کنید و به جای هر bit عدد ۱ را قرار دهید.

?

```
1*215+1*214+1*213 + 1*212 + 1*211 +
1*210 + 1*29 + 1*28 + 1*27 + 1*26 + 1*25 + 1*24 + 1*23
+ 1*22 + 1*21 + 1*20
= 32768 + 16384 + 8192 + 4096 + 2048 + 1024 + 512 +
256 + 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1
= 65535
```

برای به دست آوردن حداکثر مقدار یا رقم مبنای ۱۶ (hexadecimal)که در یک واژه قابلیت ذخیره شدن را داشته باشد، تمام گروه های ۴ bit ای را با f یا F جایگزین کنید.

1111	1111	1111	1111
f	f	f	f
= 0xffff = 0xFFFF			

= 0Xffff

= 0XFFFF

short integers

یک واژه(word) ، که متشکل از ۱۶ bit همجوار یا ۲ byte می باشد، می تواند یک عدد طبیعی (natural number)را در خود جای دهد. همان طور که قبلاً ذکر شد، بیشترین مقدار عددی (numeric value) که در یک واژه می توان ذخیره کرد ۶۵۵۳۵ است. به منظور تعریف متغیری برای این مقدار، می توان کلیدواژه ی var را به کاربرد و متغیر نام برده را با مقداری از -۳۲۷۶۸ تا 32767مقداردهی اولیه کرد. به مثال زیر توجه کنید.

?

```
1
2class Exercise
3{
4            static void Main()
5            {
6                var schoolEffective = 1400; // Number of Students
7                System.Console.Write("School Effective: ");
8                System.Console.WriteLine(schoolEffective);
9            }
        }
```

نتیجه ی زیر حاصل می گردد.

?

```
1       School Effective: 1400
2   Press any key to continue…
```

به این خاطر که byte تنها گنجایش کاراکترها و ارقام کوچک را دارد، هر زمان قصد داشتید عددی را در برنامه ی خود به کار ببرید، توجه داشته باشید که کوچکترین نمودی که می توان به کاربرد واژه است .

عدد طبیعی (natural number) را integer نیز می گویند .کوچکترین (integer) را تنها با کمک کلیدواژه ی short می توان در یک واژه ذخیره کرد. به دلیل این که integer short به صورت پیش فرض علامت دار می باشد، می تواند مقداری که از -۳۲۷۶۸ تا ۳۲۷۶۷ متغیر است را در خود ذخیره کند. در زیر نمونه ی برنامه ای که دو integer short را به کار می برد مشاهده می کنید.

?

```
1       class Exercise
2{
3            static void Main()
            {
```

```
        short numberOfPages;
        short temperature;
        numberOfPages = 842;
        temperature   = -1544;
        System.Console.Write("Number of Pages of the book: ");
        System.Console.WriteLine(numberOfPages);
        System.Console.Write("Temperature to reach during the experiment:
");
        System.Console.Write(temperature);
        System.Console.WriteLine(" degrees\n");
            }
        }
```

نتیجه ی زیر به دست می آید.

?

```
        Number of Pages of the book: 842
Temperature to reach during the experiment: -1544 degrees
```

(به دلیل این که short integer ها می توانند ارقام (number)بزرگتر از byte علامت دار را در خود جای دهند)، هر مقداری که برای byteعلامت دار تعریف می کنید را می توان برای متغیر short هم تعریف کرد.

Short integerهای بدون علامت

از متغیری که دربردارنده ی اعداد مثبت و نسبتاً کوچکی است با نام unsigned short integer یاد می شود. چنین متغیری را می توان با کلیدواژه های ushort یا var تعریف کرد short integer. ها ی بدون علامت می توانند اعدادی که در برد ۰ تا ۶۵۵۳۵ قرار دارند را در برگیرد (به همین دلیل است که در تنها ۱۶ bit جای می گیرند). مثال های آن را در زیر مشاهده می کنید.

?

```
        class Exercise
{
            static void Main()
            {
        // These variables must hold only positive integers
        ushort numberOfTracks;
        ushort musicCategory;
        numberOfTracks = 16;
        musicCategory  = 2;
        System.Console.Write("This music album contains ");
        System.Console.Write(numberOfTracks);
        System.Console.WriteLine(" tracks");
        System.Console.Write("Music Category: ");
        System.Console.Write(musicCategory);
        System.Console.WriteLine();
            }
         }
```

نتیجه ی زیر به دست می آید.

```
This music album contains 16 tracks
Music Category: 2
```

به کاربردن short integer های بدون علامت

برای به کاربردن short integer ها، فایل را به صورت زیر تغییر دهید.

```
class Order
{
    static void Main()
    {
        byte? shirts = null;
        byte? pants  = null;
        ushort? otherItems = null;
        shirts     = 4;
        pants      = 1;
        otherItems = 3;
        System.Console.WriteLine("-/- Georgetown Cleaning Services -/-");
        System.Console.WriteLine("=====================");
        System.Console.WriteLine("Item Type  Qty");
        System.Console.WriteLine("--------------------");
        System.Console.Write("Shirts      ");
        System.Console.WriteLine(shirts);
        System.Console.Write("Pants       ");
        System.Console.WriteLine(pants);
        System.Console.Write("Other Items ");
        System.Console.WriteLine(otherItems);
        System.Console.WriteLine("=====================");
        System.Console.ReadKey();
    }
}
```

2
2
2
3
2
4

به منظور اجرای برنامه، به main menu مراجعه کرده و روی Debug -> Start Debugging کلیک کنید. نتیجه ی زیر حاصل می گردد.

```
1         -/- Georgetown Cleaning Services -/-
2=========================
3Order Date: 7/15/2002
4-------------------------
5Item Type   Qty
6-------------------------
7Shirts        4
8Pants         1
9Other Items   3
10=========================
```

کلید Enter را بزنید تا پنجره ی DOS بسته شود.

حال، محیط برنامه نویسی را نیز ببندید.

هنگامی که از شما پرسیده شد، می خواهید SAVE کنید یا نه، NOرا بزنید.