

# آشنایی با ویژوال استودیو

در فصل قبلی با ساختار کلی یک برنامه C# آشنا شدید و برنامه‌های ساده را با استفاده از یک ویرایشگر، نوشته و ذخیره کردیم. سپس فایل برنامه را از طریق پنجره فرمان<sup>۱</sup> با استفاده از مترجم زبان C# (برنامه csc.exe) ترجمه کرده و فایل اجرایی EXE تولید شد. در آخر، فایل EXE را اجرا کردیم. این روش برای نوشتن و تولید برنامه‌های کوچک، خوب است اما اگر برنامه‌ای که می‌نویسیم بزرگ باشد، به کارگیری این روش کمی دشوار و زمان بر خواهد بود. به خصوص عیب‌یابی و اشکال‌زدایی آن بسیار وقت گیر است. در این فصل با برنامه‌ای آشنا خواهیم شد که همه ابزارها و لوازم مورد نیاز یک برنامه نویس در آن گردآوری شده است و کار برنامه‌نویسی را آسان می‌کند.

### پس از پایان این فصل انتظار می‌رود که فراگیر بتواند :

- ۱- IDE را تعریف کند و برای آن مثال بیاورد.
- ۲- مزایای استفاده از IDE و کاربرد VS را بیان نماید.
- ۳- یک برنامه جدید در محیط VS از نوع کنسول ایجاد کند، آن را ذخیره، ترجمه و اجرا نماید.

## ۱-۳- آشنایی با ویژوال استودیو<sup>۲</sup>

تایپ برنامه در یک ویرایشگر، ورود به پنجره فرمان، ترجمه کردن، عیب‌یابی و اشکال‌زدایی برنامه، همگی عملیاتی بودند که در فصل قبلی انجام دادید و کمی وقت گیر و پر زحمت بود، چون از یک محیط باید وارد محیط دیگری می‌شدید. برای این که راحت‌تر بتوانیم برنامه‌نویسی کنیم لازم است

<sup>۱</sup> - Command Prompt

<sup>۲</sup> - V sua Stud o

از محیطی استفاده کنیم که همه ابزارها و لوازم مورد نیاز برنامه‌نویسی در آن گردآوری و متمرکز شده باشد. به چنین محیط برنامه‌نویسی که در آن می‌توان تمام مراحل برنامه‌نویسی، ترجمه، اشکال‌یابی و سرانجام اجرا را انجام داد، IDE<sup>۱</sup> گفته می‌شود که به معنای محیط تولید برنامه متمرکز می‌باشد. یعنی همه ابزارها و امکانات لازم برای تولید برنامه در یک جا گردآوری شده است.

شرکت مایکروسافت یک IDE بسیار پیشرفته برای برنامه‌نویسی فراهم کرده است که با کمک آن می‌توان راحت‌تر برنامه بنویسیم و ترجمه و اجرا کنیم. نام این نرم افزار و ویژوال استودیو است. ویژوال استودیو یک محیط برنامه‌نویسی بسیار قوی برای تولید برنامه‌های کاربردی تحت ویندوز و بر پایه .Net Framework می‌باشد. ویژوال استودیو از چند زبان برنامه‌نویسی نظیر C#، C و VB پشتیبانی می‌کند. در این محیط علاوه بر تایپ برنامه، می‌توان برنامه را ترجمه، عیب‌یابی و سرانجام اجرا کرد. لایه نرم افزاری 4.5 NET Framework. به همراه ویژوال استودیو ۲۰۱۲ عرضه شد است. در حال حاضر<sup>۲</sup> VS 2013 عرضه شده است. خوشبختانه شرکت مایکروسافت همراه با عرضه ویژوال استودیوی تجاری، یک نسخه رایگان از این نرم افزار را تحت عنوان ویژوال استودیو اکسپرس<sup>۳</sup> نیز عرضه می‌کند که می‌توانید آن را از روی سایت شرکت مایکروسافت<sup>۴</sup> دانلود نمایید.



شکل ۱-۳- اجرای ویژوال استودیو اکسپرس ۲۰۱۲

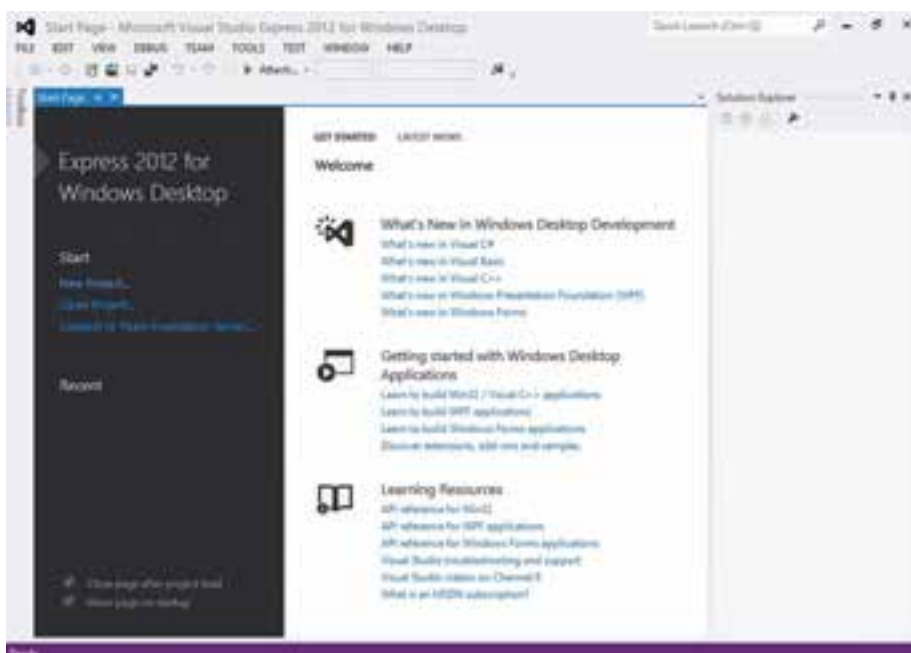
۱ \_ Integrated Deve opment Env ronnement

۲ \_ اردیبهشت ۹۳

۳ \_V sua Stud o Express

۴ \_ <http://microsoft.com/visualstudio/downloads>

ما در این کتاب با Visual Studio Express 2012 کار می‌کنیم و از این به بعد در سرتاسر کتاب از مخفف VS برای بیان کلمه ویزوال استودیو استفاده می‌کنیم. اگر نرم افزار VS 2012 را در اختیار ندارید می‌توانید از نسخه‌های قدیمی‌تر نیز استفاده کنید فقط شکل ظاهری آن ممکن است کمی متفاوت باشد.



شکل ۲-۳- پنجره آغازین ویزوال استودیو اکسپرس ۲۰۱۲

## ۲-۳- ایجاد یک پروژه جدید در ویزوال استودیو

با اجرای برنامه VS، صفحه شروع (Start Page) مطابق با شکل ۲-۳ ظاهر می‌شود، از این صفحه می‌توانید یک پروژه یا برنامه جدید (New Project...) بسازید و یا برنامه‌های قبلی خود را باز (Open Project...) کنید.

برای ایجاد یک برنامه جدید به زبان C# که در محیط کنسول کار می‌کند ابتدا روی گزینه (New Project...) در لیست سمت چپ کلیک کنید. پس از ظاهر شدن پنجره، Visual C# را انتخاب کنید (شکل ۲-۳-۳ مرحله ۱) و در وسط صفحه بر روی گزینه Console Application کلیک کنید (شکل ۲-۳-۳ مرحله ۲).



شکل ۳-۳- پنجره پروژه جدید

در قسمت Name باید نام پروژه را تایپ کنید که به صورت پیش فرض ConsoleAppliacion1 برای آن در نظر گرفته شده است. مناسب است نامی مطابق با هدف برنامه‌ای که می‌نویسید انتخاب کنید چون این نام به هیچ وجه گویا و روشن نیست.

همان‌طور که در شکل ۳-۴ مشاهده می‌کنید، در قسمت Location مسیر ذخیره پروژه نشان داده شده است. هنگامی که VS را نصب می‌کنید در داخل My Documents یک پوشه به نام Visual Studio ساخته می‌شود که در داخل آن نیز یک پوشه فرعی به نام Projects ایجاد می‌شود. در داخل این پوشه، پروژه‌های شما به‌طور پیش فرض ذخیره می‌شود. اگر بخواهید می‌توانید پروژه خود را در مسیر دیگری ذخیره کنید. مسیر دلخواه خود را با کلیک بر روی دکمه Browse مشخص کنید.

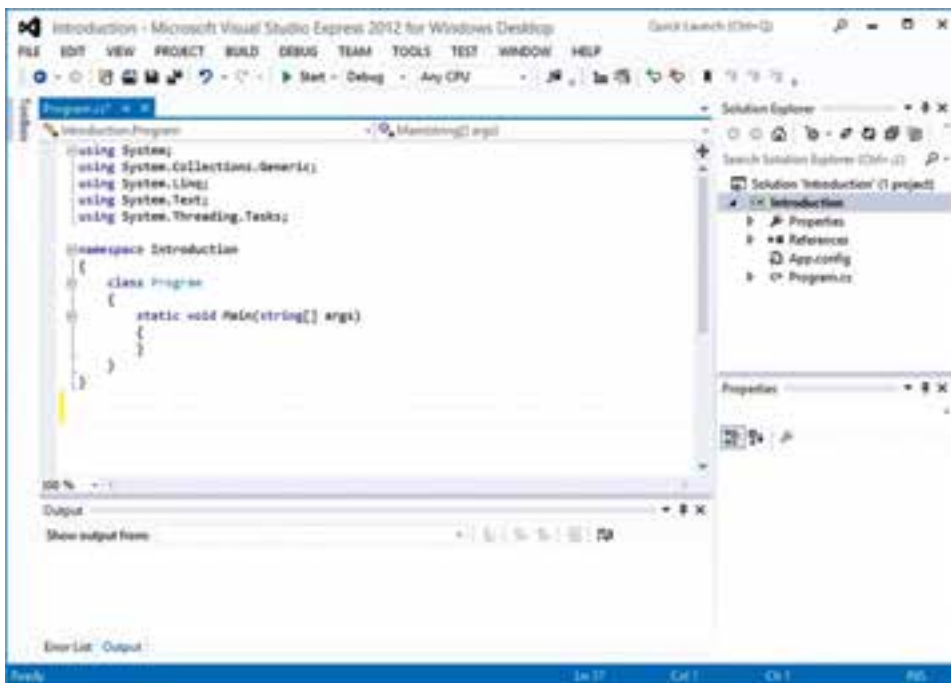


شکل ۳-۴- تعیین نام برای یک پروژه جدید

بعد از مشخص کردن محل پروژه، باید نامی برای Solution name در نظر بگیرید که نام پوشه‌ای را معین می‌کند که فایل‌های مربوط به یک یا چند پروژه در آن نگهداری می‌شود. معمولاً نام Solution با نام پروژه یکسان انتخاب می‌شود.

در حال حاضر چون هدف ما آشنایی با VS است نام Introduction را در قسمت Name می‌نویسیم و این نام برای Solution نیز انتخاب می‌شود. حال پس از تعیین نام‌ها، بر روی کلید Ok کلیک کنید. در این صورت یک پوشه با نام مذکور در مسیر پیش فرض ساخته می‌شود. درون این پوشه چندین فایل و یک پوشه فرعی با نامی که برای پروژه انتخاب کردید (در مثال ما Introduction) ساخته می‌شود.

پس از چند لحظه مشاهده خواهید کرد که در IDE پنجره‌های مختلفی نشان داده می‌شود و در یکی از پنجره‌ها صورت کلی یک برنامه C# به صورت آماده مانند شکل ۳-۵ نشان داده می‌شود.

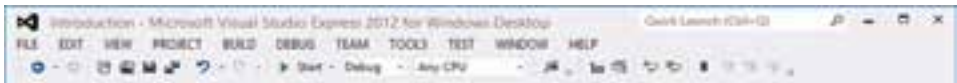


شکل ۳-۵ محیط تولید برنامه (IDE) ویژوال استودیو

قبل از این که به برنامه‌نویسی بپردازیم ابتدا به صورت مختصر با بخش‌های مختلف محیط IDE آشنا می‌شویم.

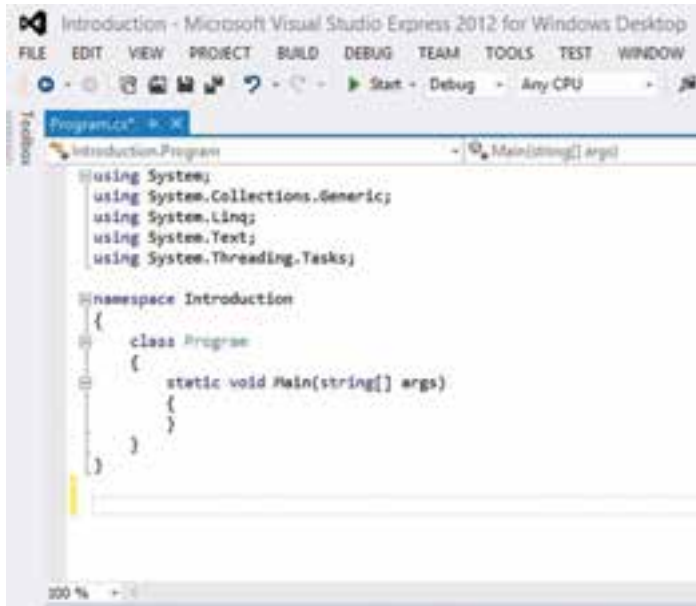
### ۳-۳-۳ معرفی بخش‌های اصلی و ابزار استودیو

۳-۳-۱ نوار منو و نوار ابزار: مانند بیشتر نرم افزارها، در قسمت بالای صفحه، منوهای مختلف VS و در زیر آن ابزارهای پرکاربرد مطابق با شکل ۳-۶ دیده می‌شود. به تدریج با این منوها و ابزارها آشنا می‌شوید.



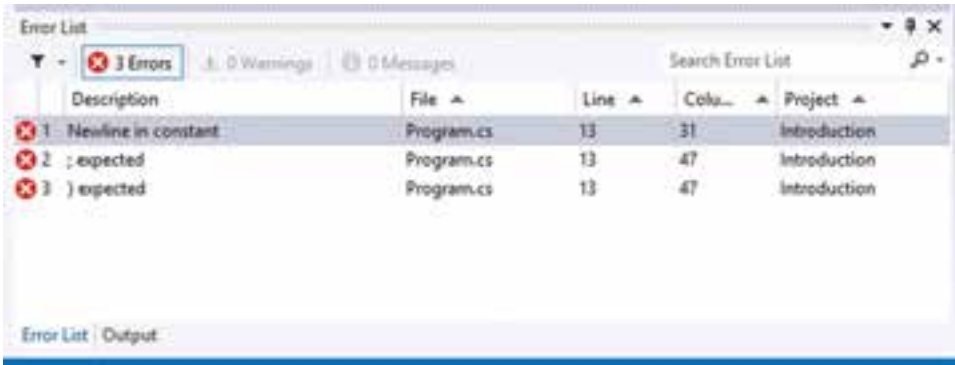
شکل ۳-۶ نوار منو و نوار ابزار محیط ویزوال استودیو

۳-۳-۲ پنجره ویرایشگر برنامه: در شکل ۳-۷ پنجره ویرایشگر برنامه نشان داده شده است. در این پنجره متن برنامه نگهداری می‌شود و می‌توانید چندین برنامه را هم زمان به صورت باز داشته باشید. در برنامه‌نویسی، این پنجره کاربرد زیاد دارد.



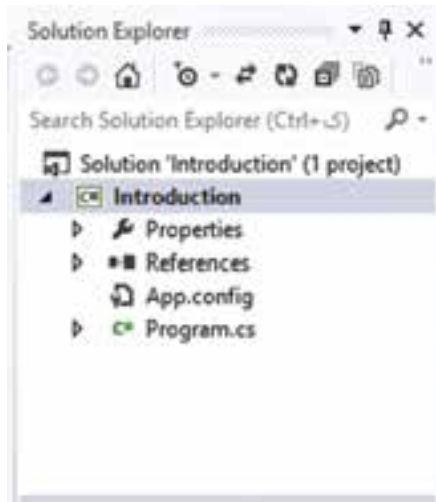
شکل ۳-۷ پنجره ویرایشگر برنامه

۳-۳-۳ پنجره لیست خطاها (Error List): در صورتی که برنامه اشکال تایپی یا ساختاری داشته باشد، خطاها و اشکالات برنامه در این پنجره مانند شکل ۳-۸ لیست می‌شوند. پس از ترجمه برنامه باید به این پنجره نگاه کنیم تا خطاهای احتمالی برنامه را متوجه شویم.



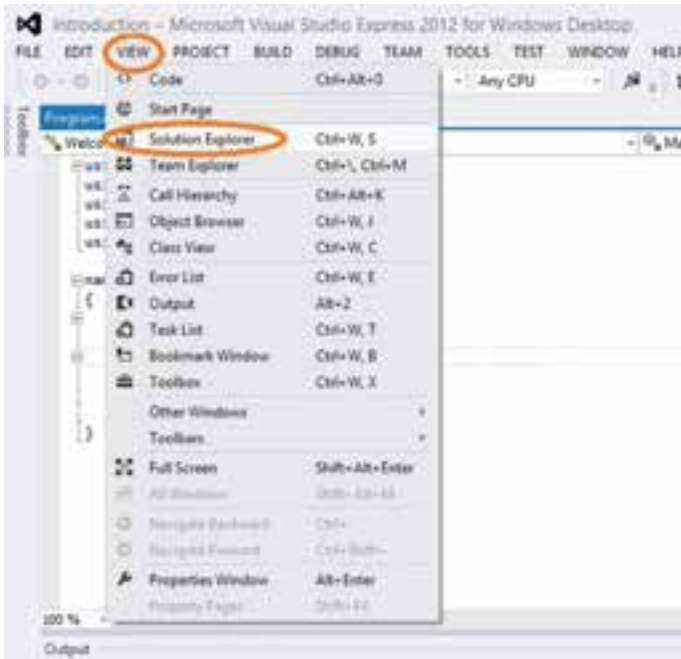
شکل ۳-۸ پنجره لیست خطاها

۳-۳-۴ پنجره (Solution Explorer): سمت راست صفحه، پنجره ای قرار دارد که ساختار پروژه و تمام فایل‌های موجود در آن را نشان می‌دهد. اگر پروژه‌ای باز نباشد، محتوای این پنجره خالی است. ما نام این پنجره را مرورگر پروژه می‌نامیم و به وسیله آن به تمام اجزای پروژه دسترسی داریم.



شکل ۳-۹ پنجره Solution Explorer

اگر پنجره مرورگر پروژه باز نیست، و آن را مشاهده نمی‌کنید از منوی View در بالای صفحه استفاده کنید. در این منو، نام تمام پنجره‌ها در شکل ۱-۳ مشاهده می‌شود، روی گزینه Solution Explorer کلیک کنید تا این پنجره دیده شود.

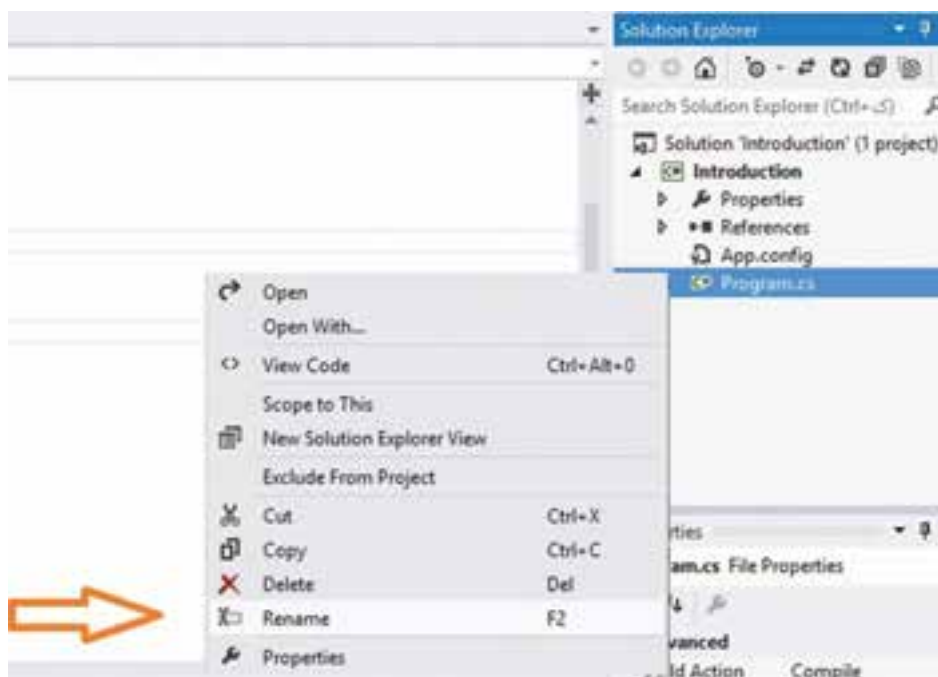


شکل ۱-۳- ظاهر کردن پنجره Solution Explorer

### ۳-۴ برنامه‌نویسی در محیط ویزوال استودیو

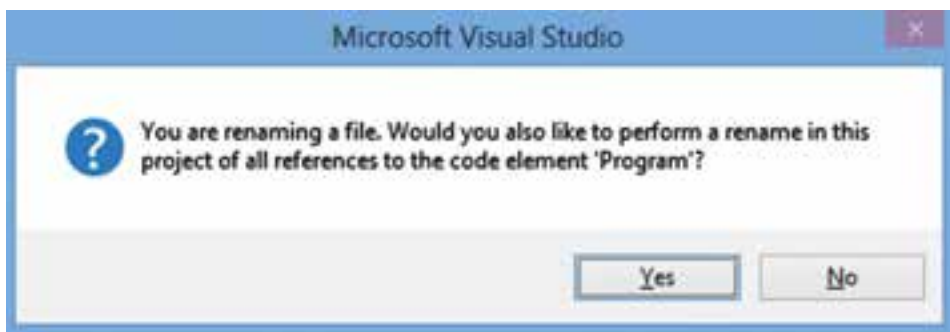
بعد از ایجاد یک پروژه جدید و وارد شدن به محیط برنامه‌نویسی (IDE)، اگر به پنجره کاوشگر Solution توجه کنید، داخل شاخه پروژه Introduction، یک فایل به نام program.cs وجود دارد. (پسوند cs. نشان‌دهنده برنامه به زبان C Sharp می‌باشد) این فایل، فایل متن برنامه است و به‌طور خودکار تولید شده است. محتوای این فایل در پنجره ویرایشگر برنامه نشان داده شده است. نام این فایل را می‌توانید مطابق با عملکرد برنامه تغییر دهید و یک نام مناسب برای آن انتخاب کنید که با دیدن نام فایل به عملکرد آن پی‌برید. در این مثال می‌خواهیم برنامه‌ای بنویسیم که یک پیام خوشامدگویی مانند فصل قبل نمایش دهد بنابراین برای تغییر نام فایل Program.cs روی آن کلیک راست کرده و نام دلخواه خود مثلاً Welcome.cs را وارد می‌کنیم (شکل ۱۱-۳).





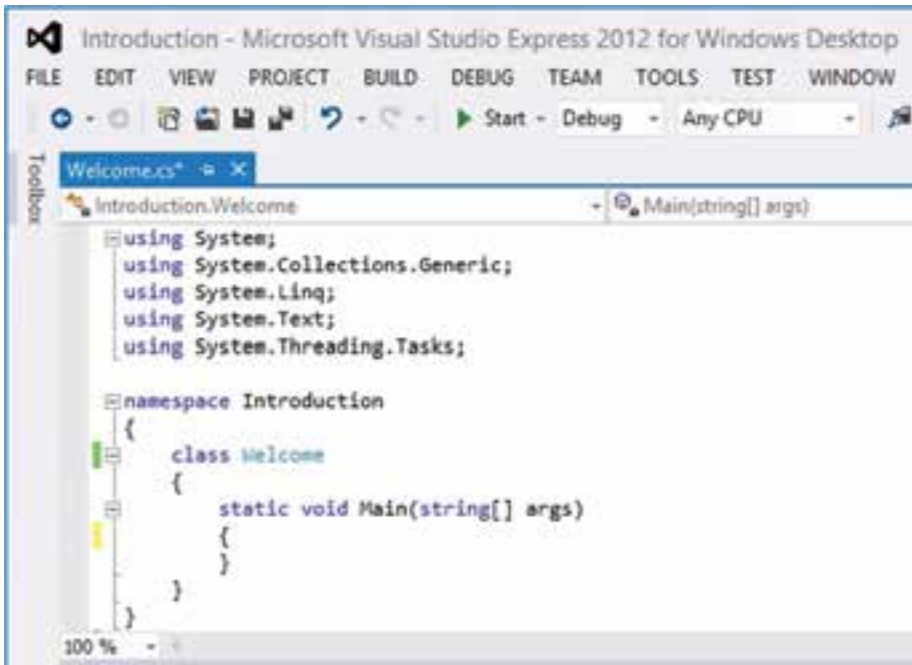
شکل ۳-۱۱- تغییر نام برنامه

پس از تغییر نام و زدن کلید Enter پنجره‌ای مانند شکل ۳-۱۲ باز می‌شود و سؤال می‌شود «آیا این تغییر نام در تمام مکان‌هایی که به این نام رجوع می‌شود نیز اعمال شود؟» اگر پاسخ مثبت یعنی Yes را انتخاب کنید مشاهده خواهید کرد که در متن برنامه، نام کلاس که قبلاً Program بود به Welcome تغییر نام می‌دهد.



شکل ۳-۱۲- تغییر نام در تمام قسمت‌هایی که به Program رجوع می‌کند

حال به پنجره کد، باز می‌گردیم، همان‌طور که در شکل ۱۳-۳ ملاحظه می‌کنید ساختار کلی یک برنامه به وسیلهٔ VS برای شما آماده می‌شود که تعدادی دستور در آن موجود است.



شکل ۱۳-۳ پنجره ویرایشگر برنامه جدید

در چند خط بالای برنامه، راهنمایی‌هایی برای مترجم یعنی دستورات using نوشته شده است و جلوی هر کدام، یک فضای نامی ذکر شده است. فضای نامی System برای شما آشنا است زیرا در فصل قبل دستور using System را در بالای برنامه نوشتیم و این کار سبب شد که استفاده و نوشتن متدهای مربوط به Console در برنامه ساده‌تر شود. در حال حاضر می‌توانید دستورات دیگر using را پاک کنید چون فعلاً به کلاسی غیر از Console نیاز نداریم.

بعد از دستورات using، دستور namespace به همراه نام پروژه (Introduction) نوشته شده است و علامت‌های آکولاد باز و بسته کل برنامه را دربرگرفته است. با دستور namespace یک فضای نامی جدید تعریف می‌شود که برای سازماندهی و دسته‌بندی پروژه‌های بزرگ مورد استفاده دارد. فعلاً به آن کار نداریم ولی لازم نیست آن را پاک کنید، بنابراین تغییری روی آن انجام ندهید.

در داخل این فضای نامی، دستور class و سپس در داخل آن متد Main نوشته شده است. نام کلاس Welcome است مگر این که در مرحله قبل نام برنامه را تغییر نداده باشید که در این صورت

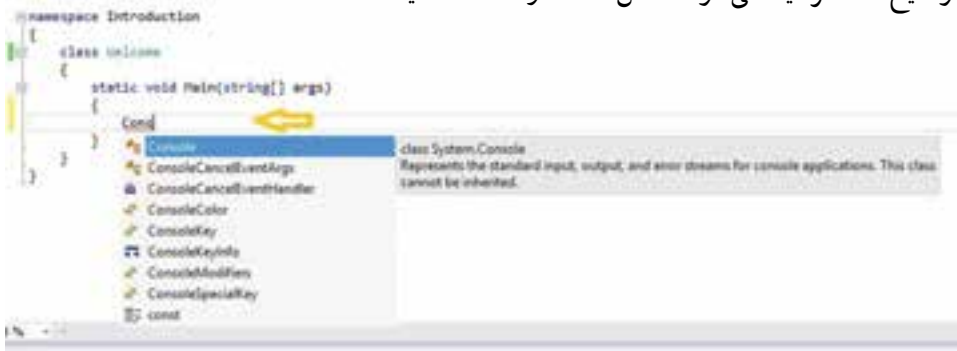
نام کلاس Program است. متد Main() نیز در داخل کلاس دیده می شود ولی داخل پرانتزهای آن عبارت string[] args نوشته شده است. می توانید این عبارت را از داخل پرانتزها پاک کنید تا برنامه ساده تر شود همان طور که در فصل قبل داخل پرانتزها خالی بود.

داخل متد Main، بین آکولادها فضا ایجاد کنید (این کار را با کلیک کردن در بین دو علامت آکولاد و سپس با زدن کلید Enter انجام دهید) تا بتوانید دستورات مورد نظر خود را بنویسید. مثلاً دستورات زیر را بنویسید :

```
Console.WriteLine("Hello World");
```

```
Console.WriteLine("Welcome to C#");
```

هنگامی که مشغول تایپ برنامه هستید باید تفاوت قابل ملاحظه ای را با روش فصل قبل که برنامه را در محیط Notepad ویندوز و یا در برنامه Notepad می نوشتید احساس کنید. اولاً کلمات با توجه به نوع آنها رنگی نوشته شده اند، مثلاً کلمات کلیدی با رنگ آبی نشان داده شده اند و ضمناً در هنگام تایپ برنامه به محض نوشتن یک حرف کلمه Console لیستی از کلمات مشابه نمایش داده می شود. با تایپ چند حرف دیگر، کلمه Console در لیست نشان داده می شود و در کنار آن یک توضیح مختصر دیده می شود. شکل ۱۴-۳ را مشاهده کنید.



شکل ۱۴-۳- کمک در تایپ دستورات در محیط ویزوال استودیو

هرگاه کلمه مورد نظر شما در لیست دیده شد و نوار آبی رنگ روی آن قرار گرفت می توانید تایپ آن کلمه را رها کنید و کلید Enter را بزنید. در این صورت کلمه به طور کامل تایپ می شود و می توانید دنباله دستورات را بنویسید. این کار باعث می شود سرعت شما در تایپ برنامه به طور چشمگیری افزایش یابد. اگر در تایپ یک دستور غلط املائی داشته باشید و یا قوانین برنامه نویسی زبان C# را رعایت نکنید در این صورت، یک خط قرمز رنگ، زیر کلمه یا مکانی که در آن اشتباه وجود دارد کشیده می شود و به شما یادآوری می کند که در آن مکان یک خطا وجود دارد و شما باید آن را برطرف کنید.

```

namespace Introduction
{
    class Welcome
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}

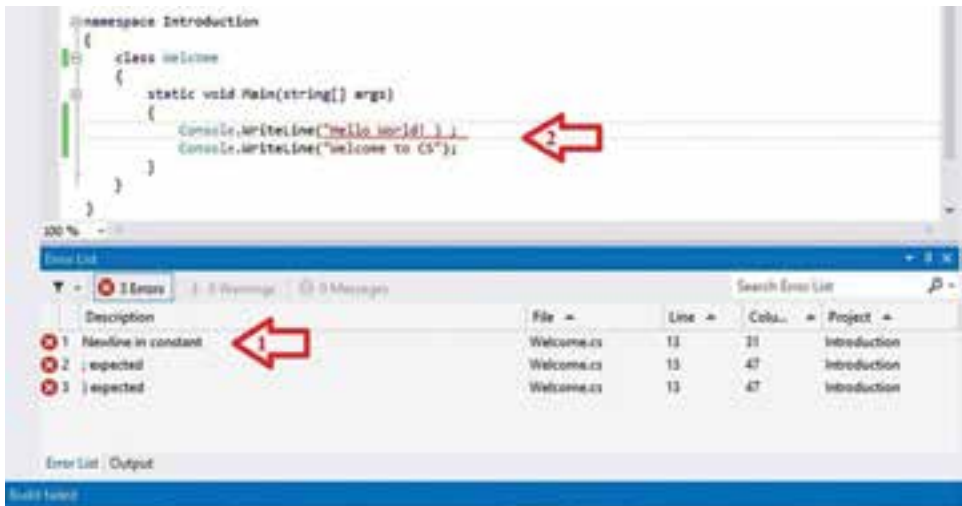
```

شکل ۳-۱۵- نشان دادن خطا در تایپ برنامه در محیط ویژوال استودیو

مثلاً در شکل ۳-۱۵ مشاهده می کنید که یک خط قرمز در زیر عبارت Hello World! کشیده شده است. با کمی دقت متوجه می شوید که علامت نقل قول انتهای پیام فراموش شده است! چون پیام ها به عنوان یک رشته باید بین علامت های نقل قول قرار داشته باشند.

### ۳-۵- ترجمه برنامه

بعد از نوشتن دستورات بالا، برای ترجمه برنامه، کلید F6 را بزنید. اگر برنامه خطا داشته باشد، خطاها در پنجره List Error دیده می شود. برای مثال شکل ۳-۱۶ خطاهای برنامه را در حالتی نشان می دهد که فراموش کرده اید انتهای رشته را با علامت " مشخص نمایید. در این شکل، پنجره Error List چند خطا را نشان می دهد؟



شکل ۳-۱۶- مراحل رفتن به محل خطا در برنامه

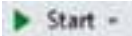
شاید این مسئله تا حدودی تعجب‌آور باشد که شما در تایپ برنامه، یک علامت نقل قول را فراموش کرده‌اید، اما در پنجره Error List، سه خطا گزارش می‌شود. بنابراین در ترجمه یک برنامه، انتظار گزارش تعداد خطای زیاد را داشته باشید. در حالت بروز خطا، باید با خواندن توضیح خطا و شماره خط برنامه که گزارش می‌شود، اقدام به رفع اشکال برنامه کنید. همواره از خطای اول شروع کنید، بنابراین در پنجره Error List، روی خطای اول دابل کلیک کنید (شکل ۳-۱۶ مرحله ۱) تا به طور مستقیم به پنجره ویرایشگر برنامه و محل خطای مزبور هدایت شوید (شکل ۳-۱۶ مرحله ۲). در این صورت اقدام به رفع اشکال کنید و سپس برنامه را دوباره ترجمه کنید. اگر خطایی گزارش شد باز سراغ خطای اول بروید و برنامه را تصحیح کنید.

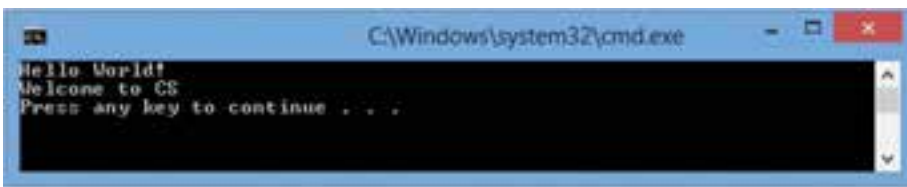
پس از رفع اشکالات برنامه اگر دوباره عمل ترجمه را انجام دهید، در پنجره خروجی (Output) پیام ترجمه بدون خطا و موفق را مانند شکل ۳-۱۷ مشاهده خواهید کرد.



شکل ۳-۱۷- پنجره خروجی در حالت بدون خطا

### ۳-۶- اجرای برنامه

بعد از ترجمه صحیح و بدون خطا، می‌توانید برنامه را اجرا کنید. برای اجرای برنامه از دکمه  بر روی نوار ابزار و یا از ترکیب کلیدی Ctrl F5 استفاده کنید. با اجرای برنامه صفحه کنسول با سه پیام مانند شکل ۳-۱۸ دیده می‌شود. دو پیام خوش آمدگویی، همان‌هایی هستند که شما نوشته‌اید. اما پیام سوم Press any key to continue... به‌طور خودکار به وسیلهٔ VS در صفحه کنسول اضافه می‌شود تا صفحه کنسول بسته نشود و شما فرصت دیدن پیام‌ها را داشته باشید. با زدن یک کلید دلخواه، پنجره کنسول بسته می‌شود زیرا دستور دیگری در برنامه وجود ندارد.



شکل ۳-۱۸- نتیجه اجرای برنامه در صفحه کنسول

## کار در کارگاه: آشنایی با محیط VS (ویژوال استودیو)

- در این قسمت، مراحل ایجاد یک پروژه، ترجمه و اجرای برنامه را تجربه خواهید کرد.
- ۱- برنامه VS را نصب کنید. (از پیوست شماره ۱ در انتهای کتاب استفاده کنید).
  - ۲- برنامه VS را اجرا کنید.
  - ۳- در صفحه آغازین، یک پروژه جدید بسازید. زبان C# را انتخاب و یک برنامه کاربردی جدید Console Application ایجاد کنید. (به توضیحات ابتدای این فصل مراجعه کنید).
  - ۴- یک نام مناسب برای پروژه انتخاب کنید. مثلاً Introduction
  - ۵- با محیط IDE ویژوال استودیو آشنا شوید و پنجره‌هایی که در این فصل معرفی شده بودند، مانند پنجره Solution Explorer را شناسایی کنید.
  - ۶- در محیط IDE، پنجره ویرایشگر برنامه را شناسایی کنید. ساختار یک برنامه باید دیده شود.
  - ۷- وارد پنجره ویرایشگر برنامه شوید و دو خط زیر را مطابق با آنچه در درس گفته شد در داخل متد Main اضافه کنید.

```
Console.WriteLine("Hello World");
```

```
Console.WriteLine("Welcome to C#");
```

- ۸- برنامه را ترجمه و سپس اجرا کنید.
- ۹- علامت نقطه ویرگول را از انتهای یکی از دستورات حذف کنید و دوباره برنامه را ترجمه کنید و به پنجره لیست خطاها دقت کنید. چند خطا گزارش شد؟
  - ۱۰- در پنجره لیست خطاها، روی توضیح اولین خطا دوبار کلیک کنید تا به ویرایشگر برنامه و محل خطا بروید و اشکال را برطرف کنید.
  - ۱۱- برنامه را ترجمه و اجرا کنید.
  - ۱۲- علامت نقل قول قبل از حرف H در پیام Hello World! را حذف کنید و دوباره برنامه را ترجمه کنید و به پنجره لیست خطاها دقت کنید. چند خطا گزارش شد؟
    - ۱۳- در پنجره لیست خطاها، روی توضیح اولین خطا دوبار کلیک کنید تا به ویرایشگر برنامه و محل خطا بروید و اشکال را برطرف کنید و سپس برنامه را ترجمه و اجرا کنید.
    - ۱۴- در داخل متد Main()، دستورات زیر را اضافه کنید.

```
static void Main(string[] args)
```

```

{
Console.BackgroundColor ConsoleColor.Blue;
Console.Clear();
Console.WriteLine("Hello World!");
Console.WriteLine("Welcome to CS");
Console.ReadKey();
}

```

۱۵- متد `Beep()` از کلاس کنسول برای ایجاد یک صدا یا صوت در برنامه استفاده می‌شود این متد را در پایان برنامه به صورت زیر اضافه کنید تا یک صوت به مدت یک ثانیه ایجاد کند.

```

Console.WriteLine("Welcome to CS");
Console.Beep();
Console.ReadKey();

```

اگر بخواهید مدت زمان نواختن صدا و همچنین فرکانس (زیر و بم) صدا را تغییر دهید، کافی است در داخل پراتز متد `Beep()`، اعدادی را به ترتیب زیر بنویسید :

```

Console.Beep ( فرکانس برحسب هرتز , مدت زمان برحسب میلی ثانیه )

```

مثلاً برای نواختن یک صدا با فرکانس ۵۰۰ هرتز به مدت ۲ ثانیه (۲۰۰۰ میلی ثانیه) دستور زیر را می‌نویسیم :

```

Console.Beep(500 , 2000 );

```

توجه داشته باشید که مقدار فرکانس را باید در محدوده مناسبی بنویسید زیرا گوش انسان فقط قادر است اصواتی با فرکانس حدود ۲۰۰ تا ۱۰۰۰۰ هرتز را خوب بشنود. صدای بم دارای فرکانس کم و صداهای ریز فرکانس بالا دارند.

۱۶- با قرار دادن اعداد مختلف به عنوان فرکانس در متد `Beep()`، محدوده شنوایی گوش خود را آزمایش کنید.

## خودآزمایی فصل سوم

- ۱- نرم افزار Visual Studio یک..... بسیار پیشرفته برای برنامه‌نویسی به زبان C# است.
- ۲- IDE مخفف کلمات..... است و بیانگر یک محیط برنامه‌نویسی است که می‌توان در آن برنامه را تایپ کرد.
- ۳- در محیط VS علاوه بر تایپ برنامه، می‌توان برنامه را.....، عیب‌یابی و سرانجام..... کرد.
- ۴- در محیط VS پس از ترجمه برنامه، خطاهای احتمالی آن در کدام پنجره دیده می‌شوند؟
- ۵- در پنجره Solution Explorer محیط VS، چه اطلاعاتی نشان داده می‌شود؟
- ۶- برنامه‌نویسی در محیط ویژوال استودیو چه برتری نسبت به استفاده از یک ویرایشگر دارد؟
- ۷- ترجمه برنامه در محیط VS چه برتری نسبت به پنجره فرمان و استفاده از مترجم CSC.EXE دارد؟
- ۸- از چه متدی برای ایجاد یک صوت استفاده می‌کنید؟ مدت زمان و فرکانس نواختن صوت را چگونه تعیین می‌کنید؟

## تمرینات برنامه‌نویسی فصل سوم

- (این تمرینات در محیط ویژوال استودیو انجام شود)
- ۱- برنامه‌ای بنویسید که نام و نام خانوادگی و تاریخ تولد شما را به زبان فارسی نمایش دهد.
  - ۲- برنامه‌ای بنویسید که ۳ صوت با فرکانس‌های ۵۰۰ و ۶۰۰ و ۷۰۰ هرتر را هر یک به مدت نیم ثانیه پشت سرهم ایجاد کند.
  - ۳- برنامه‌ای بنویسید که اطلاعات هر سطر جدول زیر را در یک خط نمایش دهد.

نام برنامه	آدرس سایت
Notepad++	<a href="http://notepad-plus-plus.org">http://notepad-plus-plus.org</a>
Visual Studio	<a href="http://microsoft.com/visualstudio/downloads">http://microsoft.com/visualstudio/downloads</a>
NET Framework 3.5	<a href="http://www.microsoft.com/en-us/download/details.aspx?id=22">http://www.microsoft.com/en-us/download/details.aspx?id=22</a>
DirectX	<a href="http://www.microsoft.com/en-us/download/details.aspx?id=35">http://www.microsoft.com/en-us/download/details.aspx?id=35</a>

- ۴- با استفاده از متدهای تغییر رنگ زمینه و رنگ قلم خروجی برنامه تمرین شماره ۳ را به صورت دلخواه رنگی کنید.



## واژگان و اصطلاحات انگلیسی فصل سوم

ردیف	واژه انگلیسی	معنی به فارسی
۱	Beep	
۲	Command Prompt	
۳	Console Application	
۴	Integrated Development Environment	
۵	Introduction	
۶	Location	
۷	New Project	
۸	Open Project	
۹	Solution Name	
۱۰	Start Page	
۱۱	Toolbar	
۱۲	Visual Studio	
۱۳	Visual Studio Express	