

کدهای سی شارپ به کدهای معادل آن در زبان میانی تبدیل می شوند.

کدهای زبان میانی در یک فایل اسمبلی ذخیره می شوند.

وقتی کدها اجرا می شوند کامپایلر JIT، کدهای زبان میانی را در لحظه به کدهایی که برای کامپیوتر قابل خواندن باشند، تبدیل می کند.

دات نت ویژگی دیگری به نام سیستم نوع مشترک (CTS) نیز دارد که بخشی از CLR است و نقشه ای است برای معادل سازی انواع داده ها در دات نت. با CTS نوع `int` در سی شارپ و نوع `Integer` در ویژوال بیسیک یکسان هستند، چون هر دو از نوع `System.Int32` مشتق می شوند. پاک کردن خانه های بلا استفاده حافظه در یک فایل (Garbage collection) یکی دیگر از ویژگیهای دات نت فریم ورک است. هنگامی که از منابعی، زیاد استفاده نشود دات نت فریم ورک حافظه استفاده شده توسط برنامه را آزاد می کند.

سی شارپ چیست؟

سی شارپ (#C) یک زبان برنامه نویسی شیء گرا است که توسط شرکت مایکروسافت ساخته شده است. سی شارپ ترکیبی از قابلیت‌های خوب ++C و Java است. اگر با این دو زبان آشنایی دارید این شانس را دارید که زبان سی شارپ را راحت یاد بگیرید. این زبان به قدری راحت است که هم کسانی که قبلاً برنامه نویسی نکرده اند و هم دانش آموزان می توانند راحت آن را یاد بگیرند. از سی شارپ می توان برای ساخت برنامه های تحت ویندوز، تحت وب، وب سرویس ها، برنامه های موبایل و بازی ها استفاده کرد. می توان به جای واژه ویژوال سی شارپ از کلمه سی شارپ استفاده کرد اما ویژوال سی شارپ به معنای استفاده همزمان از سی شارپ و محیط گرافیکی ویژوال استودیو می باشد. زبان برنامه نویسی سی شارپ تنها زبانی است که مخصوصاً برای دات نت فریم ورک طراحی شده است. سی شارپ از کتابخانه کلاس دات نت که شامل مجموعه بزرگی از اجزا از قبل ساخته شده است، استفاده می کند. این اجزا به ساخت هر چه سریعتر برنامه ها کمک می کنند. سی شارپ یک برنامه بسیار قدرتمند و شیء گرا است و با آن می توان برنامه هایی با قابلیت مدیریت بیشتر و درک آسان ایجاد کرد. ساختار این زبان نسبت به زبانهای دیگر بسیار آسان و قابل فهم است.

برای اجرای یک برنامه سی شارپ ابتدا باید دات نت فریم ورک نصب شود. سی شارپ یکی از زبانهایی است که از تکنولوژی های دیگر دات نت مانند ASP.NET, Silverlight و XNA پشتیبانی می کند. همچنین یک محیط توسعه یکپارچه دارد که آن نیز به نوبه خود دارای ابزارهای مفیدی است که به شما در کدنویسی در سی شارپ کمک می کند. این زبان به طور دائم توسط مایکروسافت به روز شده و ویژگیهای جدیدی به آن اضافه می شود. سی شارپ یکی از بهترین زبانهای برنامه نویسی دات نت است.

ویژوال استودیو

ویژوال استودیو محیط توسعه یکپارچه ای است، که دارای ابزارهایی برای کمک به شما برای توسعه برنامه های سی شارپ و دات نت می باشد. شما می توانید یک برنامه سی شارپ را با استفاده از برنامه **notepad** یا هر برنامه ویرایشگر متن دیگر بنویسید و با استفاده از کامپایلر سی شارپ از آن استفاده کنید، اما این کار بسیار سخت است چون اگر برنامه شما دارای خطا باشد، خطایابی آن سخت می شود. توجه کنید که کلمه ویژوال استودیو هم به ویژوال استودیو و هم به ویژوال سی شارپ اشاره دارد. توصیه می کنیم که از محیط ویژوال استودیو برای ساخت برنامه استفاده کنید چون این محیط دارای ویژگی های زیادی برای کمک به شما جهت توسعه برنامه های سی شارپ می باشد. تعداد زیادی از پردازش ها که وقت شما را هدر می دهند به صورت خودکار توسط ویژوال استودیو انجام می شوند. یکی از این ویژگی ها اینتل لایسنس (**Intellisense**) است که شما را در تایپ سریع کدهایتان کمک می کند. یکی دیگر از ویژگیهای اضافه شده **break point** است که به شما اجازه می دهد در طول اجرای برنامه مقادیر موجود در متغیرها را چک کنید. ویژوال استودیو برنامه شما را خطایابی می کند و حتی خطاهای کوچک (مانند بزرگ یا کوچک نوشتن حروف) را برطرف می کند، همچنین دارای ابزارهای طراحی برای ساخت یک رابط گرافیکی است که بدون ویژوال استودیو برای ساخت همچنین رابط گرافیکی باید کدهای زیادی نوشت. با این برنامه های قدرتمند بازدهی شما افزایش می یابد و در وقت شما با وجود این ویژگیهای شگفت انگیز صرفه جویی می شود. در حال حاضر آخرین نسخه ویژوال استودیو **Visual Studio 2015** است. این نسخه به دو نسخه **Visual Studio Professional** (ارزان قیمت) و **Visual Studio Enterprise** (گرانقیمت) تقسیم می شود و دارای ویژگی های متفاوتی هستند. خبر خوب برای توسعه دهندگان نرم افزار این است که مایکروسافت تصمیم دارد که ویژوال استودیو را به صورت متن باز ارائه دهد. یکی از نسخه های ویژوال استودیو، **Visual Studio Community** می باشد که آزاد است و می توان آن را دانلود و از آن استفاده کرد. این برنامه ویژگیهای کافی را برای شروع برنامه نویسی **C#** در اختیار شما قرار می دهد. این نسخه (**Community**) کامل نیست و خلاصه شده نسخه اصلی است. به هر حال استفاده از **Visual Studio Community** که جایگزین **Visual Studio Express** شده و به نوعی همان نسخه **Visual Studio Professional** است، برای انجام تمرینات این کتاب کافی است.

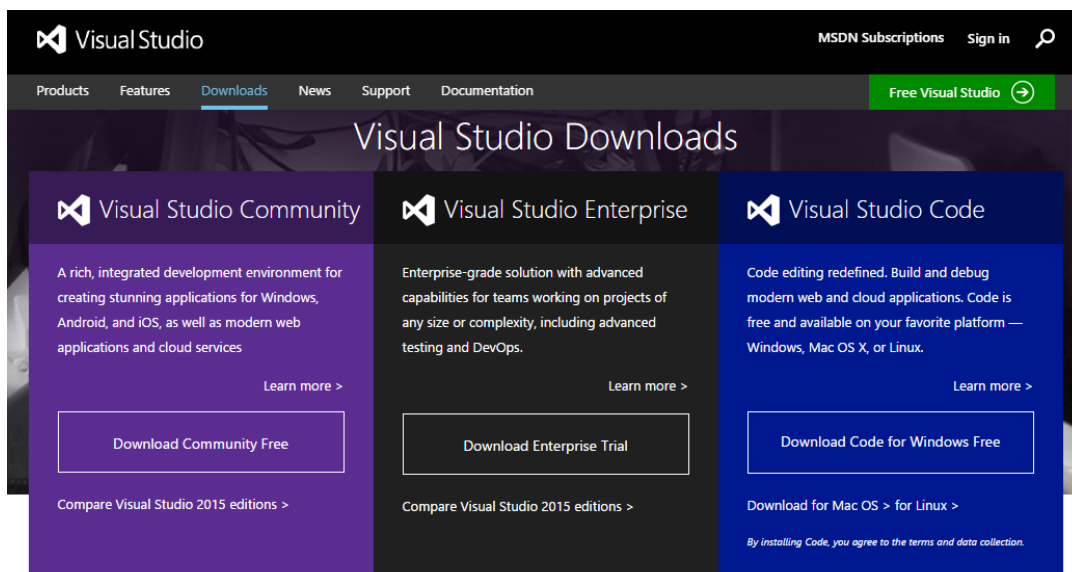
Visual Studio Enterprise 2015 دارای محیطی کاملتر و ابزارهای بیشتری جهت عیب یابی و رسم نمودارهای مختلف است که در **Visual Studio Community** وجود ندارند. ویژوال استودیو فقط به سی شارپ خلاصه نمی شود و دارای زبانهای برنامه نویسی دیگری از جمله ویژوال بیسیک نیز می باشد. رابط کاربری سی شارپ و ویژوال استودیو بسیار شبیه هم است و ما در این کتاب بیشتر تمرینات را با استفاده از سی شارپ انجام می دهیم.

دانلود و نصب ویژوال استودیو

در این درس می خواهیم نحوه دانلود و نصب Visual Studio Community 2015 را آموزش دهیم.

دانلود Visual Studio Community 2015

Visual Studio Community 2015 به صورت آزاد در دسترس است و می توانید آن را از لینک زیر دانلود کنید:
<https://www.visualstudio.com/en-us/downloads/download-visual-studio-vs.aspx>
 با کلیک بر روی لینک بالا صفحه ای به صورت زیر ظاهر می شود که در داخل این صفحه می توان با کلیک بر روی Visual Studio Community 2015 آن را دانلود کرد:



Visual Studio downloads

- [Visual Studio 2015](#)
- [Visual Studio Code](#)
- [Top Third-Party Extensions](#)
- [Team Foundation Server 2015](#)
- [Tools for Visual Studio 2015](#)
- [Visual Studio 2013](#)
- [Team Foundation Server 2013](#)
- [Tools for Visual Studio 2013](#)

Click the download category in the navigation to the left.

Then, choose the product or tool you want to view in the pane to the right.



بعد از کلیک بر روی **Visual Studio Community 2015** زبان مورد نظر و سپس فرمت **ISO** را انتخاب کرده تا تمام فایل های لازم جهت نصب **Visual Studio Community 2015** دانلود شوند. با این کار اگر می توان همین فایل **ISO** را بر روی سیستم های دیگر بدون نیاز به دانلود مجدد فایل های لازم، نصب کرد. این فایل **ISO** حدود 5 گیگابایت حجم دارد و برای دانلود آن یز به یک اینترنت پر سرعت دارید.

Visual Studio downloads

Visual Studio 2015

- Community 2015
- Enterprise 2015
- Professional 2015
- Test Professional 2015
- Express 2015 for Desktop
- Express 2015 for Web
- Express 2015 for Windows 10

Visual Studio Code

Top Third-Party Extensions

Team Foundation Server 2015

Tools for Visual Studio 2015

Visual Studio Community 2015 - Free

Visual Studio Community 2015 is a free, full-featured IDE with powerful coding productivity features, cross-platform mobile development tools for Windows, iOS and Android, tools for web and cloud development, and access to thousands of extensions. This edition of Visual Studio is available for free for individual developers, open source development, academic research, education and small professional teams. After installation, check out the Visual Studio Gallery. It provides quick access to tools, controls, and templates to help you get the most out of Visual Studio.

[Release notes](#)
[System requirements](#)
[Compatibility](#)
[SHA-1 values](#)
[Compare Visual Studio editions](#)

Choose language: English

Choose format: ISO

Download [How to install files offline](#)

با کلیک بر روی دکمه **Download**، دانلود برنامه شروع می شود.

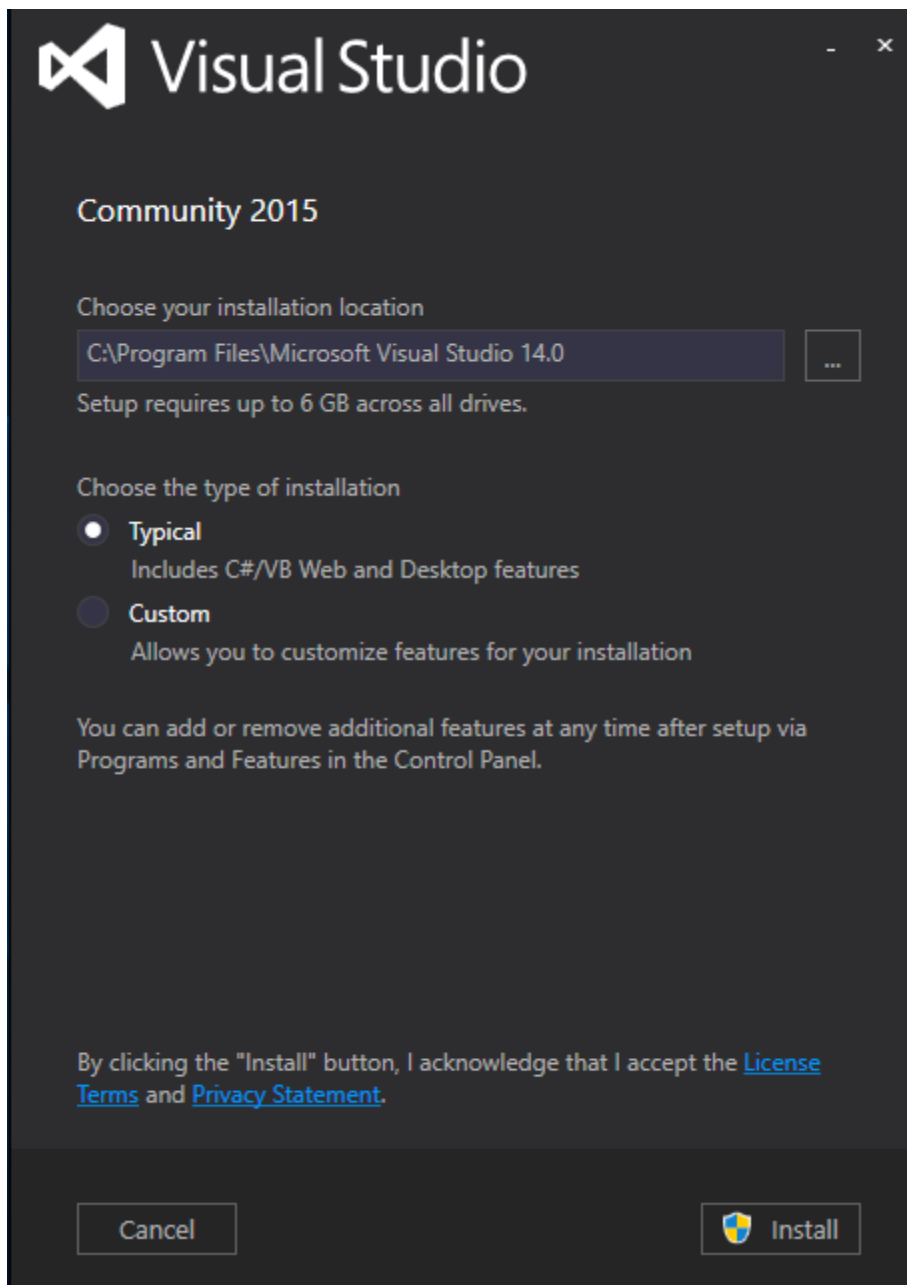
نصب Visual Studio Community 2015

بعد از دانلود فایل، شما باید قادر به باز کردن آن در ویندوز 8 یا بالاتر باشید. بعد از اجرای فایل **ISO**، به داخل آن رفته و بر روی فایل **vs_community.exe** دو بار کلیک کنید:

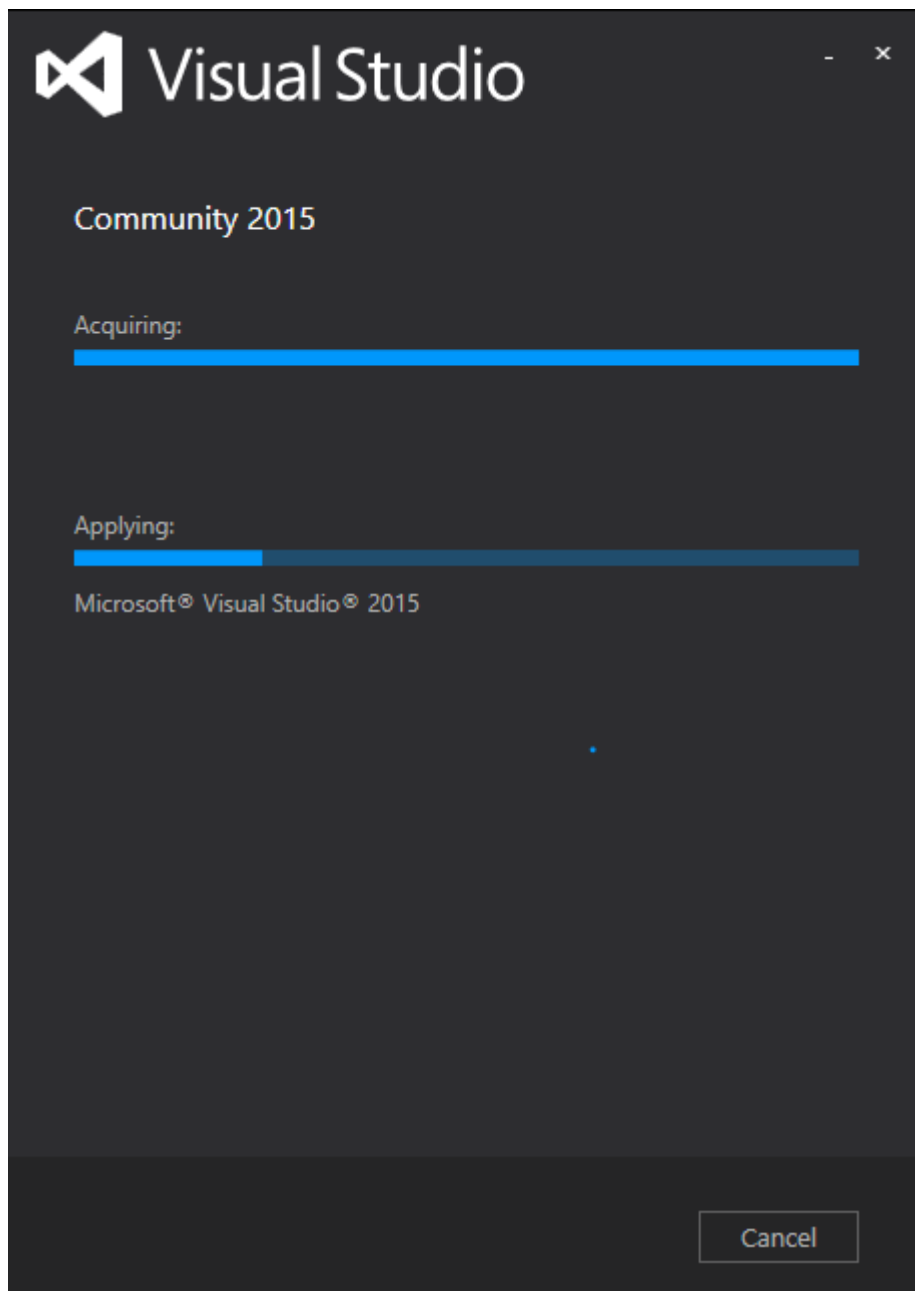
Files Currently on the Disc (4)

packages	7/9/2015 6:42 PM	File folder	
Standalone Profiler	7/9/2015 6:42 PM	File folder	
autorun	7/9/2015 5:51 PM	Setup Information	1 KB
vs_community	7/9/2015 6:42 PM	Application	2,969 KB

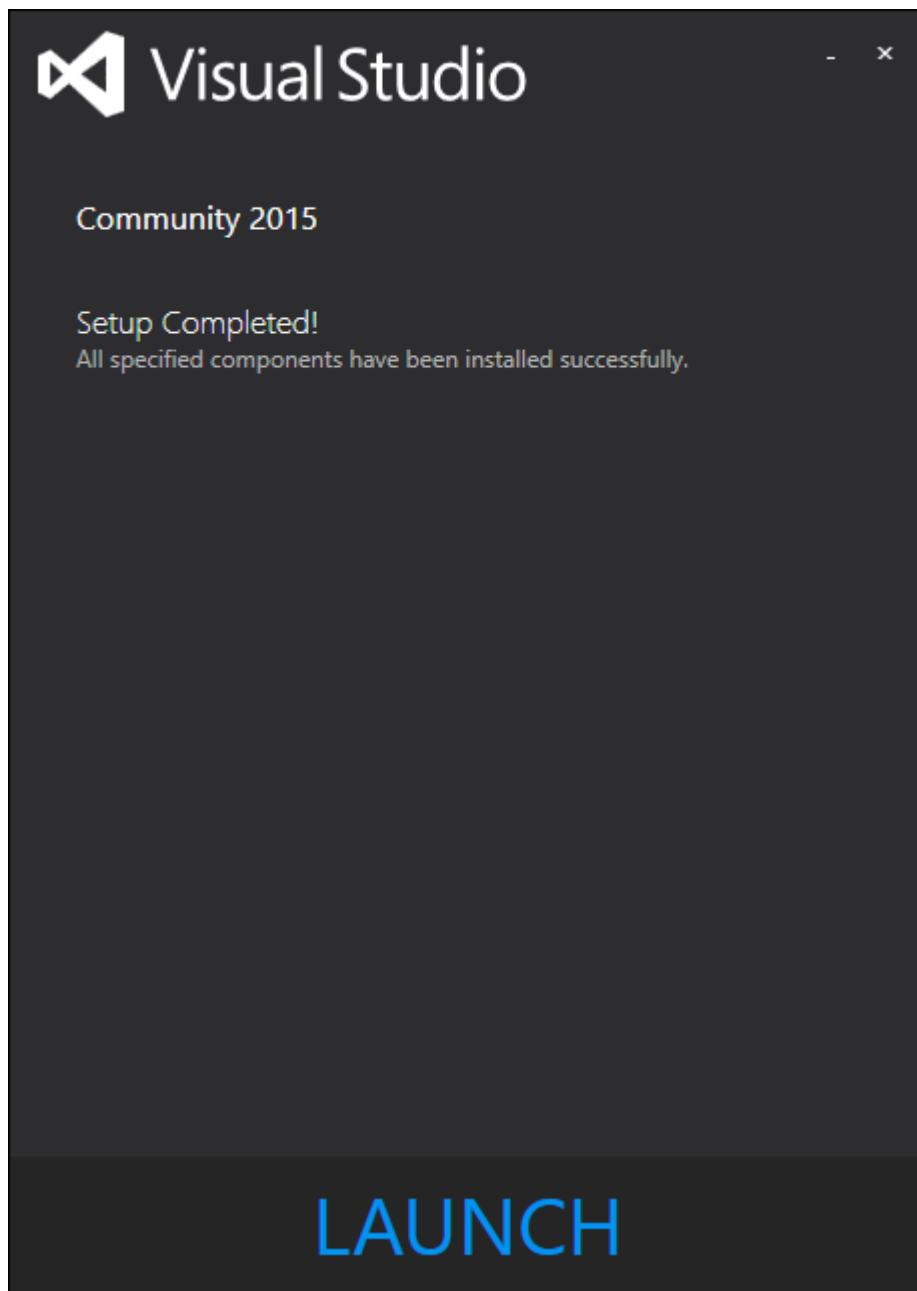
با کلیک بر روی فایل بالا صفحه ای به صورت زیر ظاهر می شود. گزینه **Typical** را انتخاب و بر روی دکمه **Install** کلیک کنید:



صبر کنید تا مرحله نصب به پایان برسد.

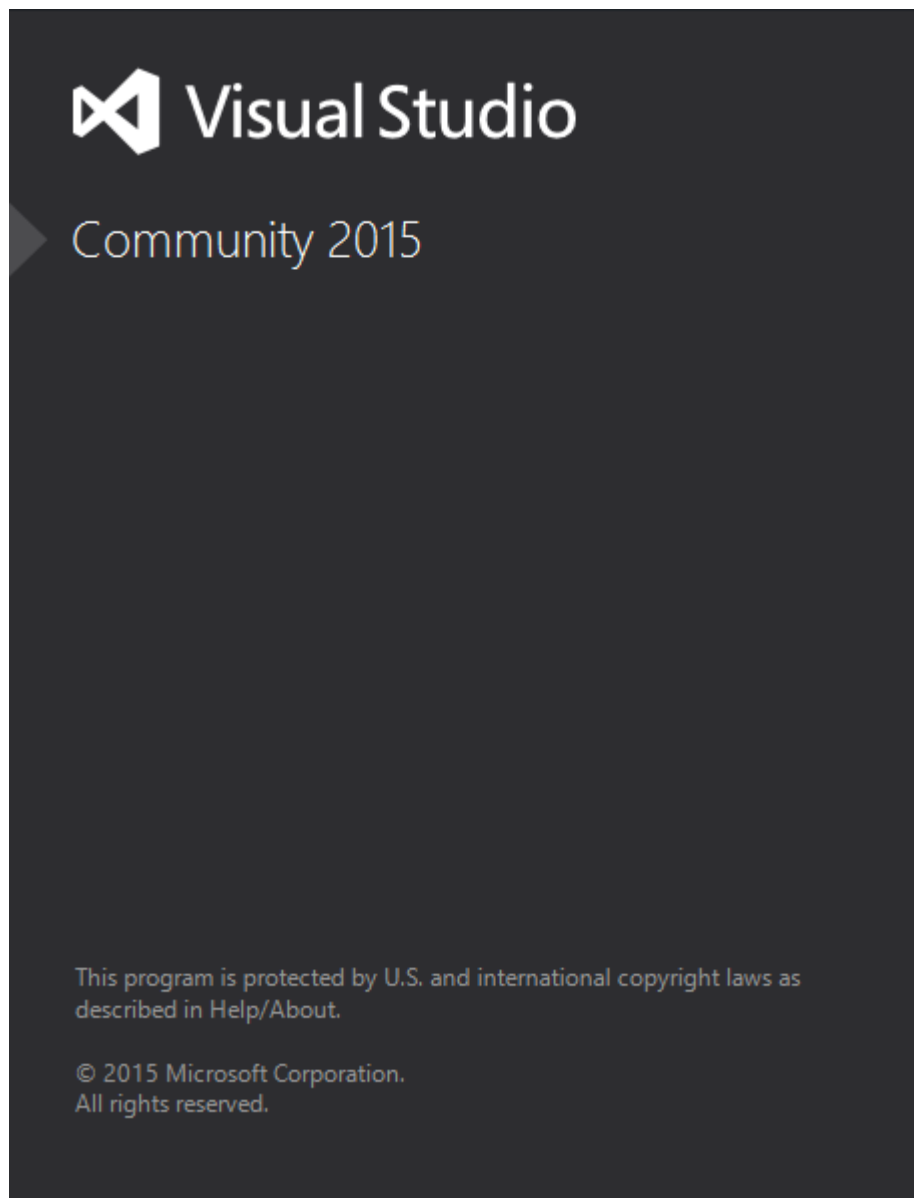


بعد از اتمام نصب، صفحه ای به صورت زیر نمایان می شود. در این صفحه بر روی دکمه **Launch** کلیک نمایید تا **Visual Studio Community** اجرا شود.

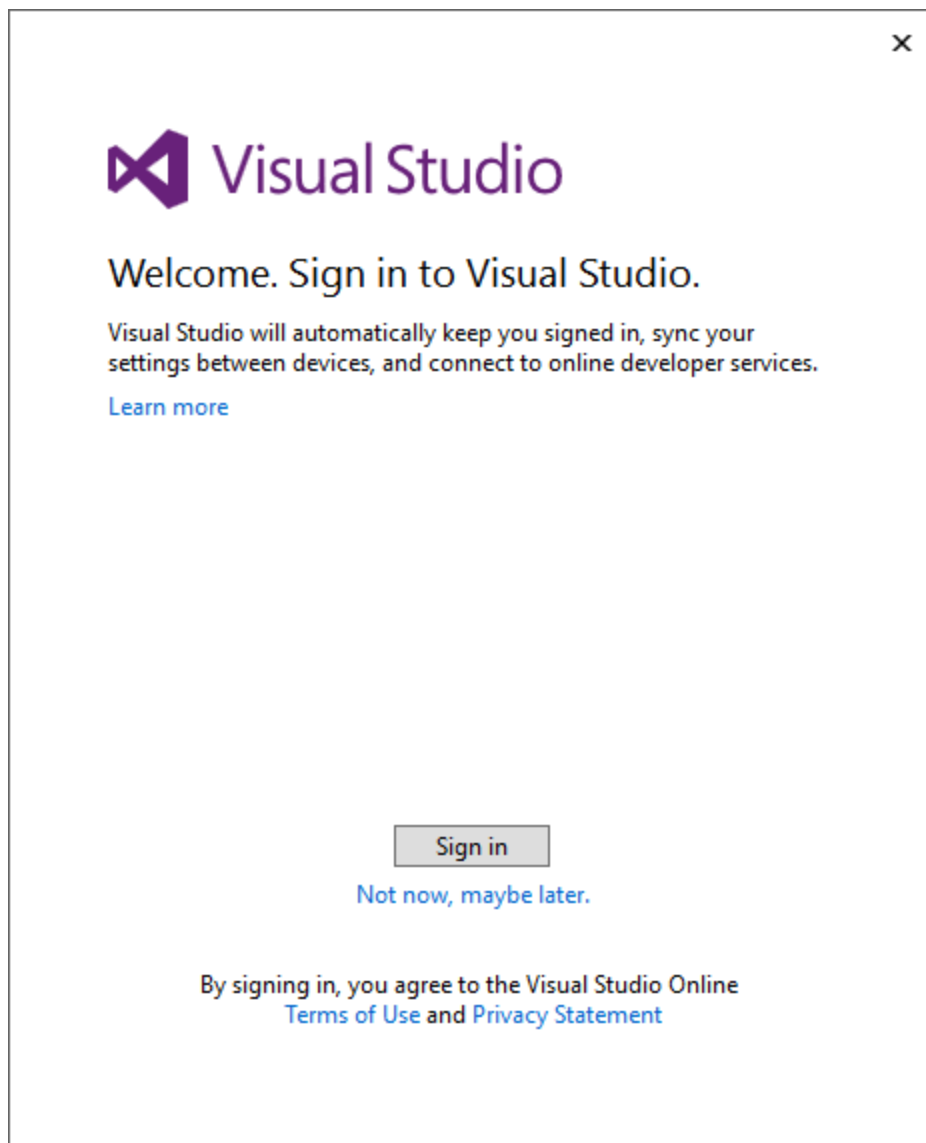


شروع کار با Visual Studio Community

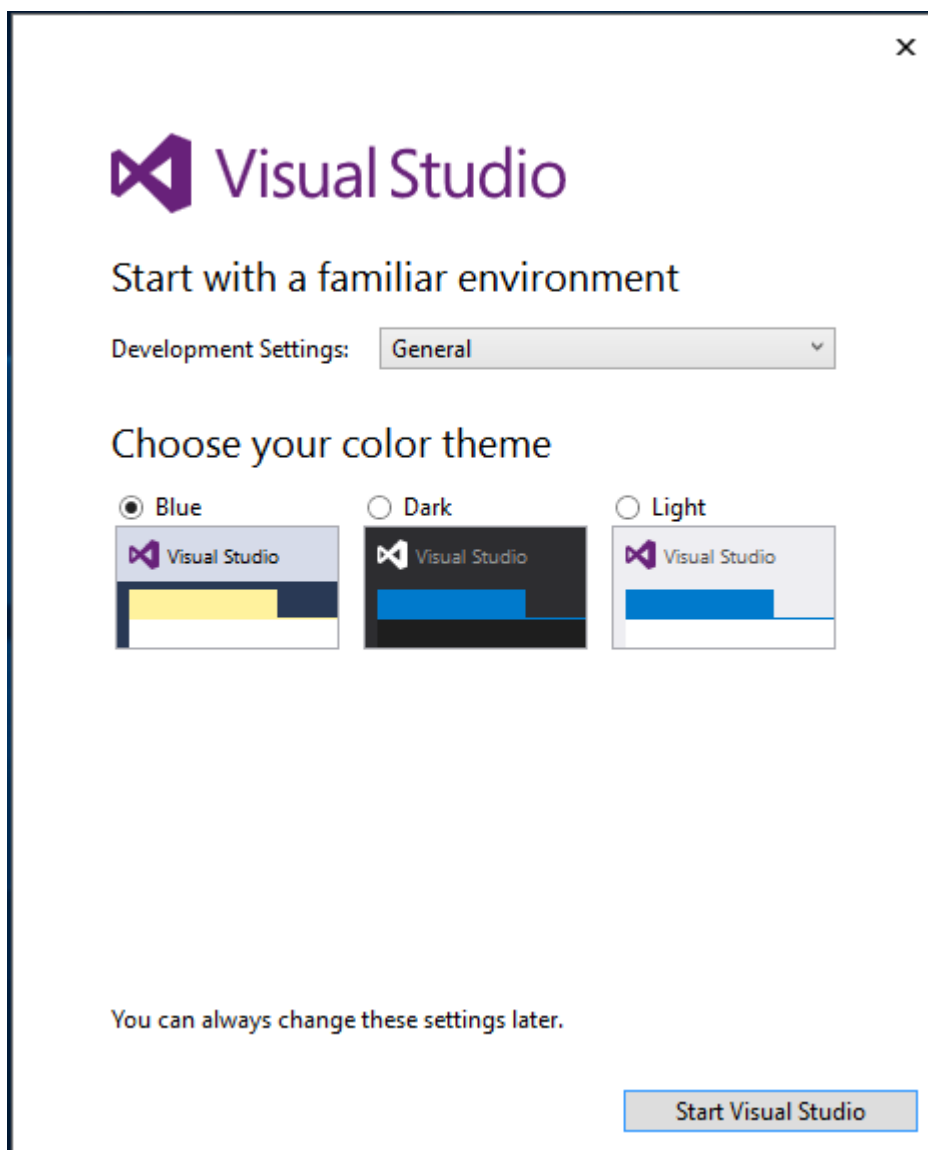
برنامه ویژوال استودیو را اجرا کرده و منتظر بمانید تا صفحه آن بارگذاری شود:



اگر دارای یک اکانت مایکروسافت باشید می توانید تغییراتی که در ویژوال استودیو می دهید را در فضای ابری ذخیره کرده و اگر آن را در کامپیوتر دیگر نصب کنید، می توانید با وارد شده به اکانت خود، تغییرات را به صورت خودکار بر روی ویژوال استودیویی که تازه نصب شده اعمال کنید. البته می توانید این مرحله را با زدن دکمه **Not now, maybe later** رد کنید:



شما می توانید از بین سه ظاهر از پیش تعریف شده در ویژوال استودیو یکی را انتخاب کنید. من به صورت پیشفرض ظاهر Blue را انتخاب می کنم ولی شما می توانید بسته به سلیقه خود، ظاهر دیگر را انتخاب کنید:



بعد از زدن دکمه **Start Visual Studio** صفحه ای به صورت زیر ظاهر می شود:

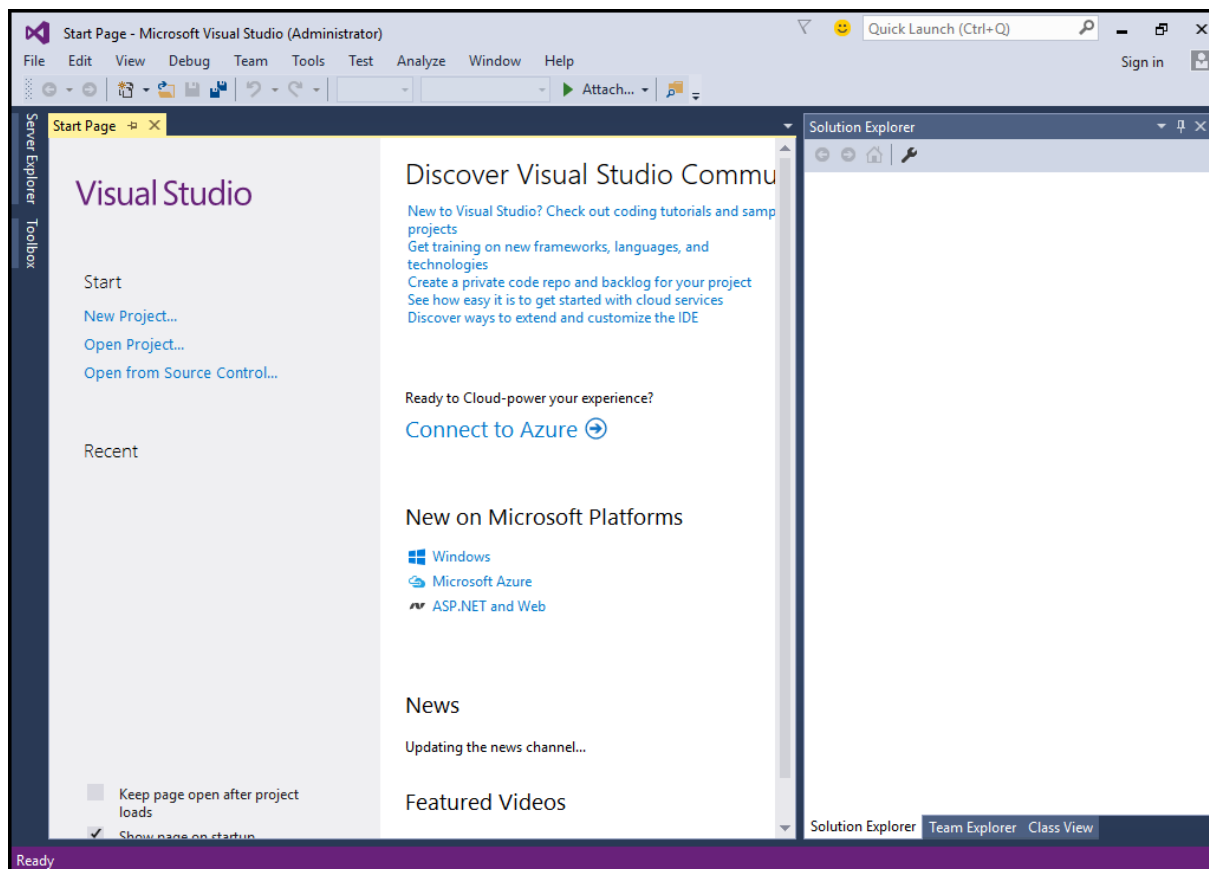


We're preparing for first use

This may take a few minutes.

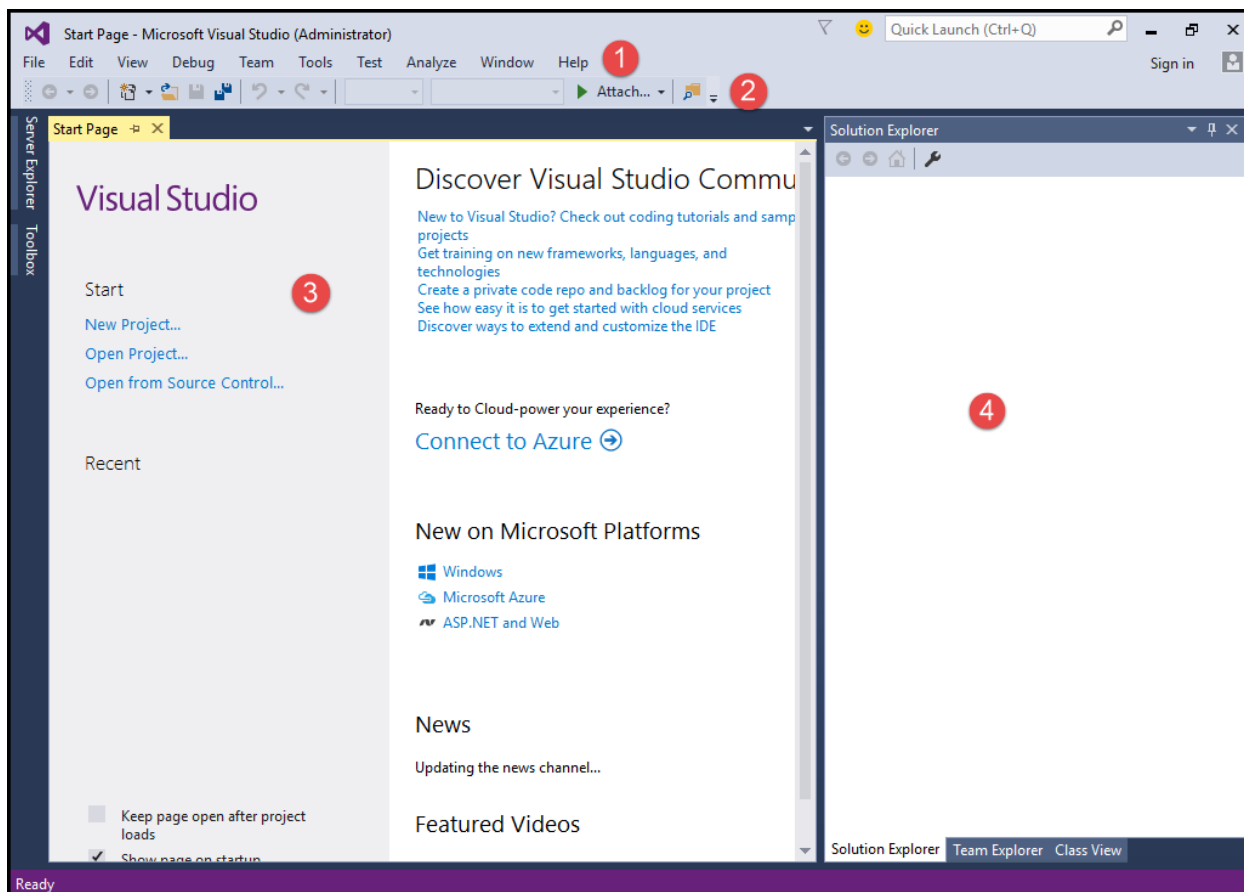


بعد از بارگذاری کامل **Visual Studio Community** صفحه اصلی برنامه به صورت زیر نمایش داده می شود که نشان از نصب کامل آن دارد:



به ویژوال استودیو خوش آمدید

در این بخش می خواهیم در باره قسمت های مختلف محیط ویژوال استودیو به شما مطالبی آموزش دهیم. لازم است که با انواع ابزارها و ویژگیهای این محیط آشنا شوید. برنامه ویژوال سی شارپ را اجرا کنید. بعد از اینکه صفحه بالا بسته شد وارد صفحه آغازین ویژوال سی شارپ می شویم:



این صفحه بر طبق عناوین خاصی طبقه بندی شده که در مورد آنها توضیح خواهیم داد.

منو بار (Menu Bar)

منو بار (1) که شامل منوهای مختلفی برای ساخت، توسعه، نگهداری، خطایابی و اجرای برنامه ها است. با کلیک بر روی هر منو دیگر منوهای وابسته به آن ظاهر می شوند. به این نکته توجه کنید که منو بار دارای آیتم های مختلفی است که فقط در شرایط خاصی ظاهر می شوند. به عنوان مثال آیتم های منوی **Project** در صورتی نشان داده خواهند شد که پروژه فعال باشد. در زیر برخی از ویژگیهای منوها آمده است:

منو	توضیح
File	شامل دستوراتی برای ساخت پروژه یا فایل، باز کردن و ذخیره پروژه ها و خروج از آنها می باشد
Edit	شامل دستوراتی جهت ویرایش از قبیل کپی کردن، جایگزینی و پیدا کردن یک مورد خاص می باشد
View	به شما اجازه می دهد تا پنجره های بیشتری باز کرده و یا به آیتم های toolbar آیتمی اضافه کنید.
Project	شامل دستوراتی در مورد پروژه ای است که شما بر روی آن کار می کنید.

Debug	به شما اجازه کامپایل، اشکال زدایی و اجرای برنامه را می دهد
Data	شامل دستوراتی برای اتصال به دیتابیس ها می باشد.
Format	شامل دستوراتی جهت مرتب کردن اجزای گرافیکی در محیط گرافیکی برنامه می باشد.
Tools	شامل ابزارهای مختلف، تنظیمات و... برای ویژوال سی شارپ و ویژوال استودیو می باشد.
Window	به شما اجازه تنظیمات ظاهری پنجره ها را می دهد.
Help	شامل اطلاعاتی در مورد برنامه و ویژوال استودیو می باشد

The Toolbars

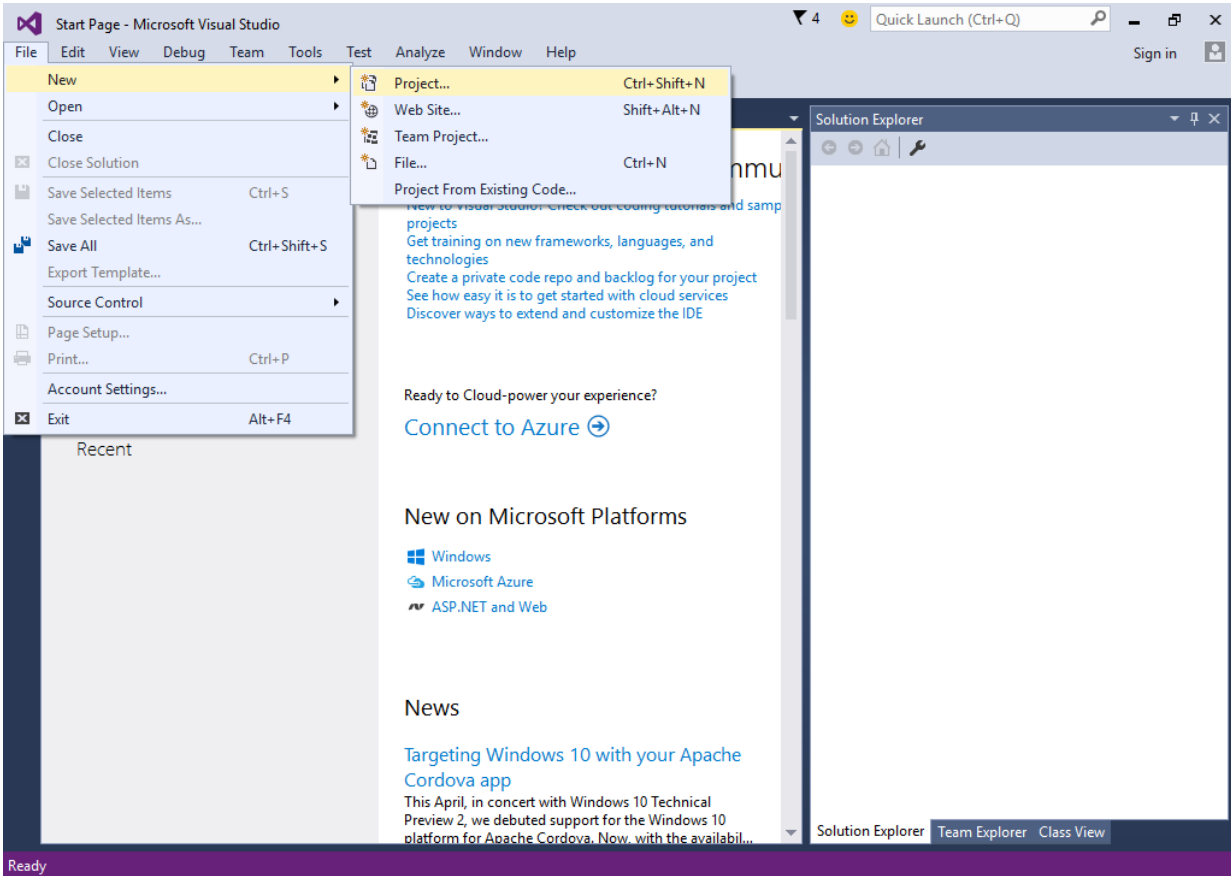
Toolbar (2) به طور معمول شامل همان دستوراتی است که در داخل منو ها قرار دارند. **Toolbar** همانند یک میانبر عمل می کند. هر دکمه در **Toolbar** دارای آیکنی است که کاربرد آنرا نشان می دهد. اگر در مورد عملکرد هر کدام از این دکمه ها شک داشتید می توانید با نشانگر موس بر روی آن مکث کوتاهی بکنید تا کاربرد آن به صورت یک پیام (**tool tip**) نشان داده شود. برخی از دستورات مخفی هستند و تحت شرایط خاص ظاهر می شوند. همچنین می توانید با کلیک راست بر روی منطقه خالی از **toolbar** و یا از مسیر **View > Toolbars** دستورات بیشتری به آن اضافه کنید. برخی از دکمه ها دارای فلش های کوچکی هستند که با کلیک بر روی آنها دیگر دستورات وابسته به آنها ظاهر می شوند. سمت چپ هر **toolbar** به شما اجازه جا به جایی آن را می دهد.

صفحه آغازین (Start Page)

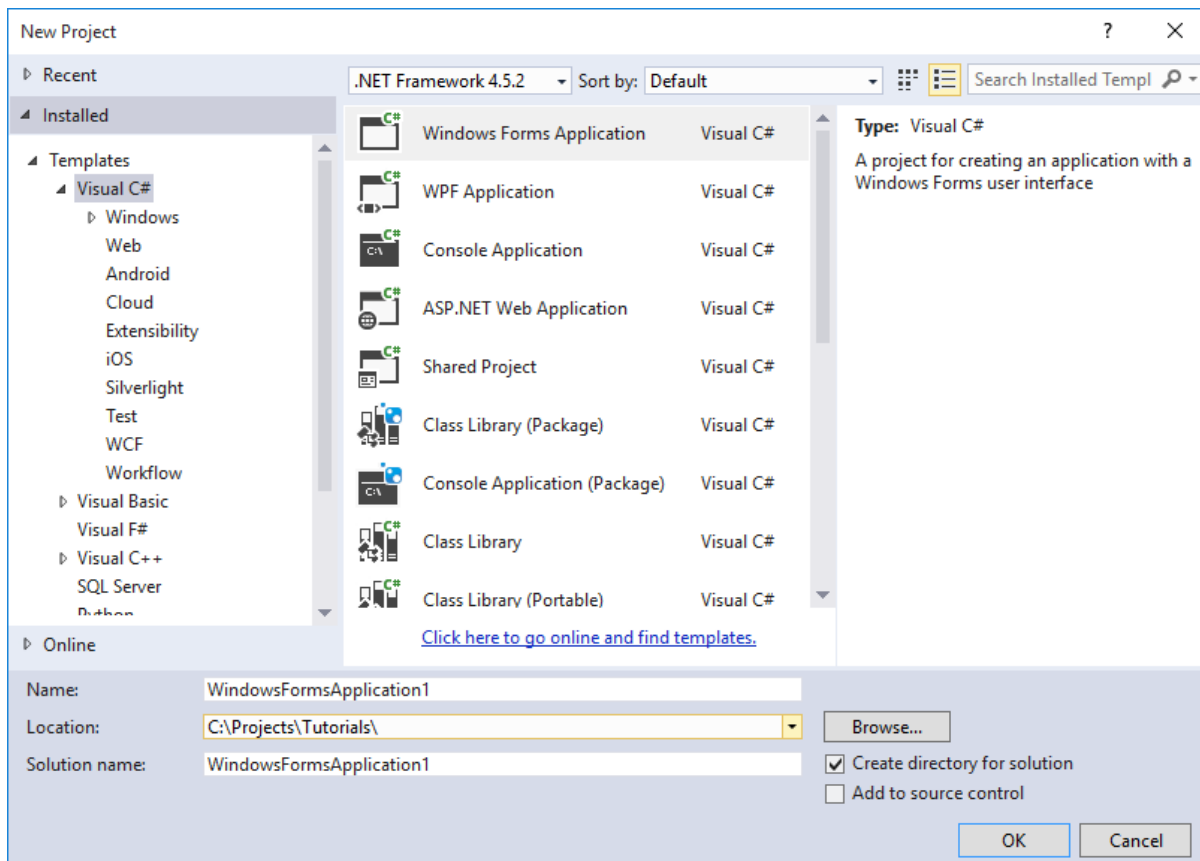
برای ایجاد یک پروژه و باز کردن آن از این قسمت استفاده می شود. همچنین اگر از قبل پروژه ای ایجاد کرده اید می توانید آن را در **Recent Projects** مشاهده و اجرا کنید. بخشهای مهم ویژوال سی شارپ توضیح داده شد در مورد بخشهای بعدی در درسهای آینده توضیحات بیشتری خواهیم داد.

گردشی در ویژوال استودیو

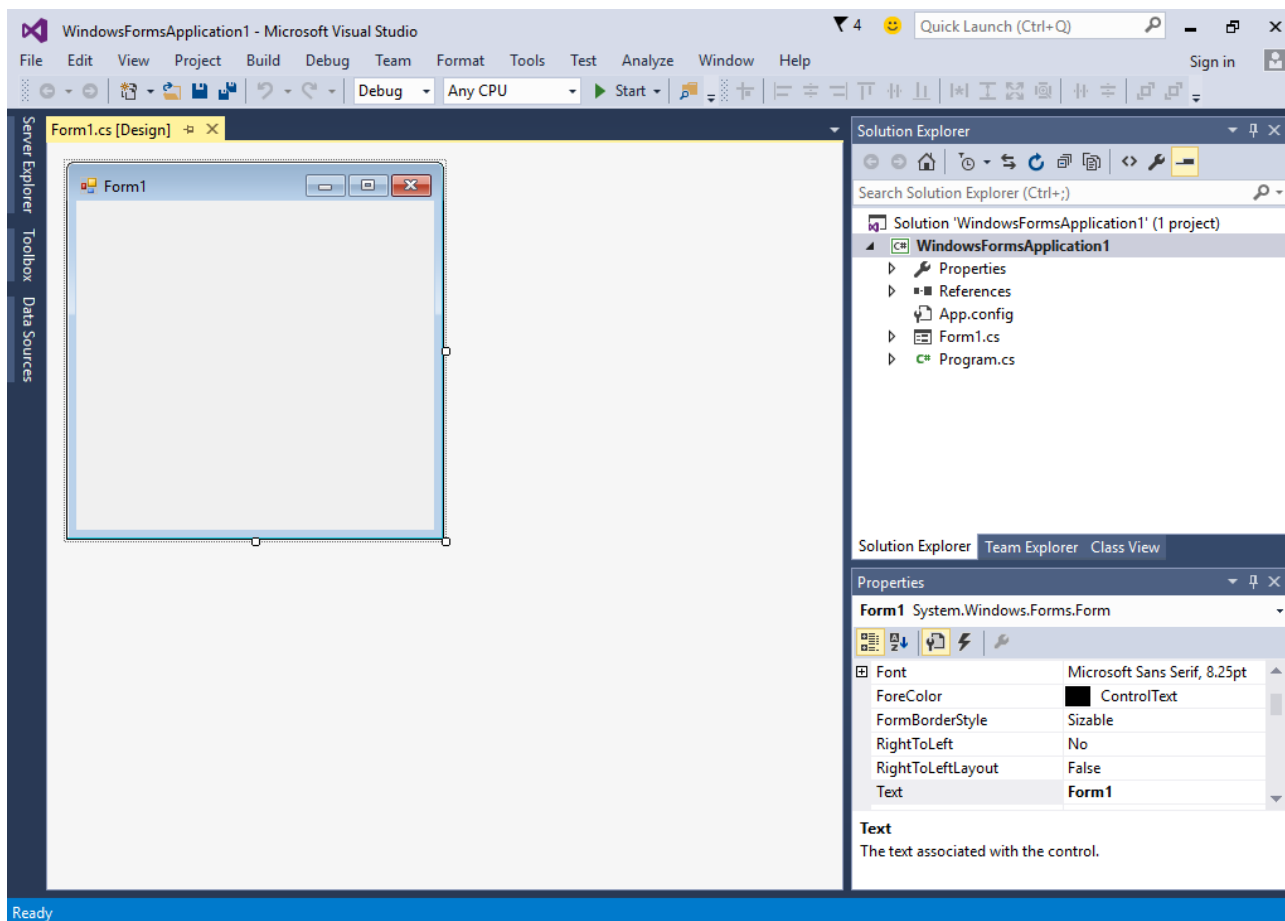
Visual Studio Community از تعداد زیادی پنجره و منو تشکیل شده است که هر کدام برای انجام کار خاصی به کار می روند. اجازه دهید با نفوذ بیشتر در محیط ویژوال استودیو با این قسمتها آشنا شویم. از مسیر **File > New Project** یک پنجره فرم ایجاد کنید.



پنجره ای به شکل زیر نمایش داده خواهد شد.



همانطور که در شکل بالا نشان داده شده است گزینه **Windows Forms Application** و یک اسم برای پروژه انتخاب می کنیم و بر روی دکمه **OK** کلیک می کنیم تا صفحه زیر نمایان شود:



مشخصات فرم بالا عبارت است از:

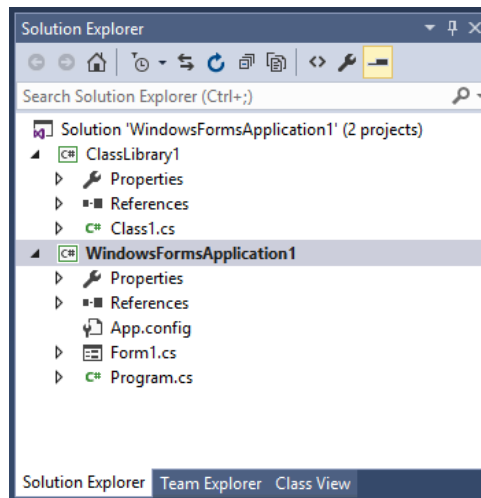
1 - صفحه طراحی (Design)

این صفحه در حکم یک ناحیه برای طراحی فرم های ویندوزی شما است. فرم های ویندوزی رابطهای گرافیکی بین کاربر و کامپیوتر هستند و محیط ویندوز نمونه بارزی از یک رابط گرافیکی یا GUI است. شما در این صفحه می توانید کنترلهایی مانند دکمه ها ، برچسب ها و ... به فرمتان اضافه کنید. جزئیات بیشتر در مورد فرمهای ویندوزی و کنترلها و برنامه نویسی شیء گرا در فصل فرم های ویندوزی آمده است. اما توصیه می شود ابتدا مبانی برنامه نویسی را مطالعه کنید.

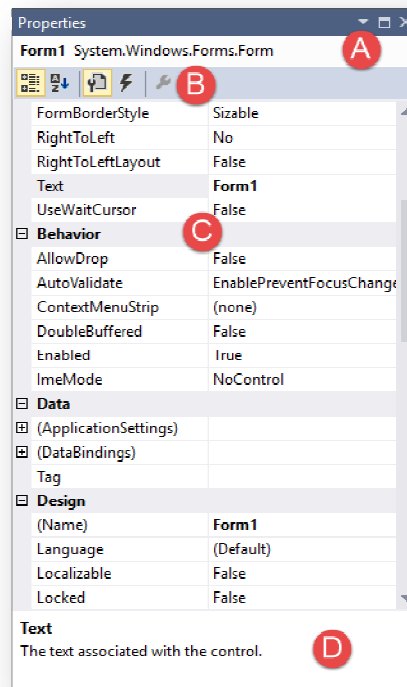
2 - مرورگر پروژه (Solution Explorer)

پروژه و فایل های مربوط به آن را نشان می دهد. یک Solution برنامه ای که توسط شما ساخته شده است را نشان می دهد. ممکن است این برنامه یک پروژه ساده یا یک پروژه چند بخشی باشد. اگر Solution Explorer در صفحه شما نمایش داده

نمی شود می توانید از مسیر **View > Other Windows > Solution Explorer** و یا با کلیدهای میانبر **Ctrl+Alt+L** آنرا نمایان کنید. اگر چندین پروژه در حال اجرا هستند پروژه ای که با خط برجسته (**Bold**) نشان داده شده پروژه فعال می باشد و هنگام اجرای برنامه اجرا می شود. اگر بخواهید پروژه ای را که فعال نیست اجرا کنید، بر روی **Solution Explorer** کلیک راست کنید و سپس گزینه **Set as StartUp Project** را انتخاب نمایید. شکل زیر یک **Solution** با دو پروژه را نشان می دهد. هر پروژه شامل فایلها و فولدرهای مربوط به خود است.



3 - پنجره خواص (Properties)



پنجره خواص (Properties)، خواص و رویدادهای مختلف هر آیتم انتخاب شده اعم از فرم، فایل، پروژه و کنترل را نشان می دهد. اگر این پنجره مخفی است، می توانید از مسیر **View > Properties Window** یا کلید میانبر **F4** آنرا ظاهر کنید. در مورد خواص در درسهای آینده مفصل توضیح خواهیم داد. خاصیت ها، ویژگیها و صفات اشیا را نشان می دهند. به عنوان مثال یک ماشین دارای خواصی مانند رنگ، سرعت، اندازه و مدل است. اگر یک فرم یا کنترل را در صفحه طراحی و یا یک پروژه یا فایل را در **Solution Explorer** انتخاب کنید پنجره خواص مربوط به آنها نمایش داده خواهد شد. این پنجره همچنین دارای رویدادهای مربوط به فرم یا کنترل انتخاب شده می باشد. یک رویداد (event) اتفاقی است که در شرایط خاصی پیش می آید مانند وقتی که بر روی دکمه (button) کلیک و یا متنی را در داخل جعبه متن (text box) اصلاح می کنیم. کمبو باکس (combo box) شکل بالا که با حرف **A** نشان داده شده است به شما اجازه می دهد که شیء مورد نظرتان (دکمه، فرم و...) را که می خواهید خواص آنرا تغییر دهید انتخاب کنید. این کار زمانی مفید است که کنترلهای روی فرم بسیار کوچک یا به هم نزدیک بوده و انتخاب آنها سخت باشد. در زیر کمبو باکس بالا دکمه های مفیدی قرار دارند. **(B)** برخی از این دکمه ها در شرایط خاصی فعال می شوند. دکمه اول خاصیت اشیا را بر اساس دسته های مختلفی مرتب می کند. دومین دکمه خواص را بر اساس حروف الفبا مرتب می کند که پیشنهاد می کنیم از این دکمه برای دسترسی سریع به خاصیت مورد نظرتان استفاده کنید. سومین دکمه هم وقتی ظاهر می شود که یک کنترل یا یک فرم را در محیط طراحی انتخاب کنیم. این دکمه به شما اجازه دسترسی به خواص فرم ویا کنترل انتخاب شده را می دهد. چهارمین دکمه (که به شکل یک رعد و برق نمایش داده شده) رویدادهای فرم ویا کنترل انتخاب شده را می دهد. در پایین شکل بالا توضیحات کوتاهی در مورد خاصیت ها و رویدادها نشان داده می شود. بخش اصلی پنجره خواص **(C)** شامل خواص و رویدادها است. در ستون سمت چپ نام رویداد یا خاصیت و در ستون سمت راست مقدار آنها آمده است. در پایین پنجره خواص جعبه توضیحات **(D)** قرار دارد که توضیحاتی درباره خواص و رویدادها در آن نمایش داده می شود.

تغییر ظاهر ویزوال استودیو

اگر موقعیت پنجره ها و یا ظاهر برنامه ویزوال استودیو را دوست نداشته باشید، می توانید به دلخواه آن را تغییر دهید. برای این کار بر روی نوار عنوان (title bar) کلیک کرده و آنرا می کشید تا پنجره به شکل زیر به حالت شناور در آید:

داده‌ها در سی شارپ

`bool`: true یا false
`byte`: داده 8 بیتی بدون علامت
`short`: عدد صحیح 16 بیتی
`int`: عدد صحیح 32 بیتی
`long`: عدد صحیح 64 بیتی
`float`: عدد اعشاری 32 بیتی
`double`: عدد اعشاری 64 بیتی
`Char`: کاراکتر 16 بیتی
`string`: کاراکترهای 16 بیتی

شما می‌توانید یک داده سطح بالاتر را با داده سطح پایین تر مقداردهی کنید. (در اینجا منظور من از سطح مجموعه اعداد و حافظه متغیر می باشد)

```
float y = 7.0f; //y is of type float
int j; //j is of type int
y = j; //convert int to float
```

البته امکان مقداردهی یک داده سطح پایین تر با داده سطح بالاتر نیز وجود دارد. به این عمل اصطلاحاً Casting گفته می‌شود.

```
1)
j = (int)y; //convert float to integer
2)
float x = 1.0E45;
int k = (int) x;
3)
int k;
bool gtnum;
gtnum = (k > 6); //true if k is greater than 6
```

در نظر داشته باشید که ممکن است عمل casting با خطا مواجه شود. همچنین در نظر داشته باشید که امکان مقداردهی متغیرهای Boolean با اعداد همانند c یا ++ وجود ندارد. و تبدیل متغیر از نوع Boolean به دیگر متغیرها امکانپذیر نیست.

تبدیل انواع اعداد و رشته ها به یکدیگر

با استفاده از متد Convert می‌توانید اعداد را به رشته و رشته را به عدد تبدیل نمایید. البته متد Convert تبدیلات دیگری را نیز انجام می‌دهد که پس از تایپ دات بعد از متد Convert می‌توانید همه آنها را ببینید.

```
string s = Convert.ToString(x);  
float y = Convert.ToSingle(s);  
float x = 12.341514325f;  
string s = x.ToString("###.###"); //gives 12.342
```

تعیین نوع و مقدار برای متغیر در زمان تعریف:

```
float loan = 1.23f; //float  
long pig = 45L; //long  
int color = 0x12345; //hexadecimal
```

در سی شارپ سه مقدار ثابت از قبل تعریف شده وجود دارد که عبارتند از:

true, false, null

برای قرار دادن کارکترهای خاص غیر رشته ای (عموما کارکترهای کنترلی) از رشته کاراکترهای از پیش تعریف شده استفاده می‌شود:

```
'\n' newline (line feed)  
'\r' carriage return  
'\t' tab character  
'\b' backspace  
'\f' form feed  
'\0' null character  
'\" double quote  
'\'' single quote  
'\\' backslash
```

نامگذاری متغیر:

در نامگذاری متغیرهای دقت کنید که حروف کوچک با حروف بزرگ متفاوت هستند در نتیجه سه کلمه زیر سه متغیر مختلف را تعریف می‌کنند:

```
temperature  
Temperature  
TEMPERATURE
```

یکی از قابلیت‌های سی شارپ در مقداردهی متغیرها استفاده از چند عملگر مساوی است:

```
i = j = k = 0;
```

در آخر هر متغیری قبل از استفاده حتما باید تعریف شده باشد. شما می‌توانید تعریف متغیر را در هر قسمتی از کد خودتون قرار بدید.

اپراتورها و عملگرها

برای تعریف یک متغیر کفایست بدینصورت عمل نمایید: اول نام نوع متغیر مورد نظرتون رو تایپ کنید و بعد نامی را برای متغیر انتخاب نموده و در آخر می‌توانید اون رو مقداردهی اولیه نیر بکنید. استفاده از چند علامت مساوی در مقداردهی اولیه متغیر:

همانند C شما در #C نیز قادر به استفاده از چند علامت مساوی برای مقداردهی به متغیرها هستید. به مثال زیر دقت کنید:

```
i = j = k = 0;
```

خیلی ساده است، این دستور در زمان کامپایل برای CPU به دستوری مانند دستور زیر تبدیل می‌شود:

```
i = 0; j = 0; k = 0;
```

اپراتورهای محاسباتی:

+	جمع
-	تفریق
*	ضرب
/	تقسیم
%	باقیمانده تقسیم صحیح

اپراتورهای منطقی:

&	عمل ترکیب AND روی بیتها
	عمل ترکیب OR روی بیتها
^	عمل ترکیب یای انحصاری روی بیتها
~	متمم یک عدد باینری
>>n	حرکت بیتها به سمت راست
<<n	حرکت بیتها به سمت

همانند C++، C و جاوا شما در #C اجازه استفاده از عملگرهای کاهش و افزایش را دارید. همچنین شما می‌توانید از دستورات انتساب خلاصه شده نیز استفاده کنید:

```
// Increment and Decrement Operators i = 5;  
j = 10;
```

```
x = i++; //x = 5, then i = 6
y = --j; //y = 9 and j = 9
z = ++i; //z = 7 and i = 7
// Combining Arithmetic and Assignment Statements
x = x + 3; //can also be written as:
x += 3; //add 3 to x; store result in x
//also with the other basic operations:
temp *= 1.80; //multiple temp by 1.80
z -= 7; //subtract 7 from z
y /= 1.3; //divide y by 1.3
```

در سی شارپ برای تصمیم گیری از دستور `if` استفاده می‌شود. شما حتما باید شرط خود را داخل پرانتز قرار دهید. اگر روال اجرایی شرط شما بیش از یک دستور باشد باید از `{ }` استفاده شود. در غیر اینصورت می‌توانید پس از دستور `if` دستور مورد نظر خود را قرار دهید. اگر می‌خواهید در صورت عدم برقراری شرط شما دستور یا دستورات خاص دیگری اجرا شود باید از `else` استفاده نمایید. در این حالت اگر شرط برقرار باشد، مجموعه ای از دستورات و اگر هم شرط برقرار نباشد مجموعه دیگری از دستورات می‌تواند اجرا شود.

```
if ( y > 0 )
    z = x / y;
else
    z = 0;
Console.WriteLine("z = " + z);
```

عملگرهای مقایسه ای:

همانند تمامی زبانهای برنامه نویسی سی شارپ نیز دارای عملگرهای مقایسه ای است. اما در نوع نمایش عملگر تفاوت وجود دارد. در سی شارپ از دو مساوی برای مقایسه یکسان بودن استفاده می‌شود. اگر شما به ترکیب شرطها در یک دستور `if` نیاز داشته باشید می‌تواند از سه عملگر " و "، "یا" و نقیض استفاده کنید. البته این سه عملگر مختص دستور `if` نیستند.

>	بزرگتر
<	کوچکتر
==	مساوی
!=	مخالف
>=	بزرگتر مساوی
<=	کوچکتر مساوی

عملگرهای منطقی:

&&	عملگر AND منطقی
----	-----------------



	عملگر OR منطقی
~	عملگر NOT منطقی

```
if ( (0 < x) && ( x <= 24) )  
    Console.WriteLine("Time is up");
```

با استفاده از دستور switch شما می‌توانید مقادیر مختلف امکانپذیر رو برای یک متغیر آزمایش کنید و سپس دستور(های) مناسب رو اجرا نمایید. متغیر شما برای مقایسه باید یک متغیر عددی یا رشته ای باشد که داخل پرانتز قرار گرفته است:

```
switch ( j )  
{  
    case 12:  
        System.out.println("Noon");  
        break;  
    case 13:  
        System.out.println("1 PM");  
        break;  
    default:  
        System.out.println("some other time...");  
}
```

با استفاده از دستور break انتهای هر بخش case مشخص همیشه. برای تست متغیر با چند مقدار می‌توانید دستورات case رو نوشته و در آخرین دستور، از break استفاده کنید.

توضیحات کد در سی شارپ:

در سی شارپ دو روش برای نوشتن توضیحات وجود دارد. روش تک خطی و روش محدوده ای

```
//C# single-line comment  
/* also can go on  
for any number of lines*/
```



حلقه ها در سی شارپ

در سی شارپ چهار نوع حلقه وجود دارد که بر اساس نیاز می توان از یکی از آنها استفاده نمود. این حلقه ها عبارتند از: `while`, `do-while`, `for`, `foreach`

while: این حلقه بسیار ساده است و تا زمانی که شرط حلقه برقرار باشد، دستورات داخل حلقه اجرا می شوند.

```
i = 0;
while ( i < 100)
{
    x = x + i++;
}
```

do-while: این حلقه نیز مانند حلقه `while` می باشد با این تفاوت که حلقه `while` می تواند اصلا اجرا نشود اما حلقه `do-while` حداقل یکبار اجرا می شود

```
i = 0;
do {
    x += i++;
}
while (i < 100);
```

for: این حلقه یک حلقه ساخت یافته و دارای سه قسمت است: قسمت مقداردهی اولیه، شرط و عملگر تغییر شمارنده حلقه. این سه قسمت توسط ";" از یکدیگر جدا می شوند

```
for (i = 0; //initialize i to 0
     i < 100 ; //continue as long as i < 100
     i++) //increment i after every pass
```

در حلقه بالا شمارنده حلقه از عدد صفر شروع می شود. در هر مرحله از اجرا شمارنده `i` با عدد ۱۰۰ مقایسه می شود و پس از بررسی اگر `i` کوچکتر از ۱۰۰ باشد بدنه حلقه اجرا می شود سپس به `i` مقدار ۱ اضافه می شود و ...

تعریف متغیر در حلقه:

در صورت نیاز به تعریف متغیر می توانید شمارنده حلقه را در زمان تعریف حلقه تعریف کنید. در این حالت شمارنده حلقه تعریف شده فقط در همان حلقه قابل دسترسی است. به مثال زیر توجه کنید:

```
for (int i =0; i< 5; i++) {
    x[i] = i;
}
System.Console.WriteLine("i=" + i.ToString());
```

استفاده از شمارنده حلقه، خارج از بدنه حلقه ای که متغیر در آن تعریف شده باشد باعث تولید خطا خواهد شد. پس اجرای خط آخر مثال فوق باعث تولید خطا خواهد شد.

استفاده از کاما " ," در تعریف حلقه:

با استفاده از کاما می‌توانید بیش از یک شمارنده حلقه در تعریف حلقه داشته باشید. علاوه بر این شما می‌توانید در بخش عملگر نیز بیش از یک عملگر را استفاده نمایید:

```
for (x=0, y= 0, i =0; i < 100; i++, y +=2)
{
    x = i + y;
}
```

کد فوق را می‌توان به شکل زیر نیز نوشت:

```
x = 0;
y = 0;
for ( i = 0; i < 100; i++)
{
    x = i + y;
    y += 2;
}
```

foreach: در این حلقه شما می‌توانید عناصر یک مجموعه را دور بزنید. این مجموعه می‌تونه شما هر نوع عنصری باشد. برای درک بیشتر به مثال زیر توجه کنید:

```
float[] z = {1.0f, 2.9f, 5.6f};
ArrayList arl = new ArrayList ();
for (int j = 0; j < z.Length ; j++) {
    arl.Add (z[j]);
}
// Sultion no 1 to access all member of "arl" with for loop
for (j = 0; j < arl.Count ; j++) {
    Console.WriteLine (arl[j]);
}
// Sultion no 2 to access all member of "arl" with foreach
loop
foreach (float a in arl) {
    Console.WriteLine (a);
}
```

سی شارپ C# زبانی است که بر اساس زبانهای ++C سی پلاس پلاس، VB ویژوال بیسیک و Java جاوا پیاده سازی شده است. هر دوی زبانهای C# و VB.NET از توابع کتابخانه ای یکسان بهره می‌برند و