

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

معرفی پروژه های C# قسمت اول

مدرس : مهندس افشین رفوآ

معرفی پروژه های C# قسمت اول

از زبان برنامه نویسی برای دادن دستور انجام کار خاصی به کامپیوتر استفاده می شود. زبان برنامه نویسی مانند زبان محاوره از کلمات معنی دار، گرامر، فرمول ها و قوانین خاص تشکیل شده که برای ساختن جملات و مفاهیم معنادار باید از آن قوانین پیروی کرد.

زبان های برنامه نویسی متعددی وجود دارند. برخلاف زبان انسان ها، زبان های برنامه نویسی کامپیوتر بسته به اهداف یا نقش هایی که ایفا می کنند، طبقه بندی می شوند.

سخت افزار عبارتند از مجموعه قطعاتی که دستور را از زبان برنامه نویسی دریافت می کنند. برخی زبان های برنامه نویسی برای بخش (سخت افزاری) خاصی تعبیه شده اند و یا محیط خاصی را هدف دستورات خود قرار می دهند. چنین محیطی از سیستم عامل خاصی استفاده می کند که از یک وسیله (مثل موبایل) به وسیله ی دیگر (کامپیوتر) قابل انتقال نمی باشد. برخی زبان های برنامه نویسی هم بخش یا قطعه ی خاصی را هدف نمی گیرند.

برخی زبان ها برای رفع یک مسئله ی خاص طراحی شده اند و توانایی انجام وظایف معمول (زبان های دیگر) را ندارند. برای مثال، زبان هایی وجود دارند که فقط قادر به نشان دادن محتویات وب هستند و از آن ها نمی توان برای انجام محاسبات ریاضی و هندسی بهره گرفت. برخی برنامه ها طوری تعبیه شده اند که تنها با مقدارهای (value) خاصی مثل (true یا false) کار می کنند. تعدادی زبان برنامه نویسی هم داریم که به حل تمام مسائل می پردازند.

C# از آن دسته زبان هایی است که علاوه بر اجرای دستورات، به حل مسائل نیز می پردازد. این زبان بر روی تمام سیستم عامل ها از جمله ویندوز قابل اجرا است. **C#** از جمله زبان هایی است که در **Microsoft.NET Framework** به کار می رود. **Microsoft.NET Framework** کتابخانه ای از اشیاء است که به کمک آن می توان در کامپیوتر چیزهایی را به وجود آورد یا ترسیم کرد.

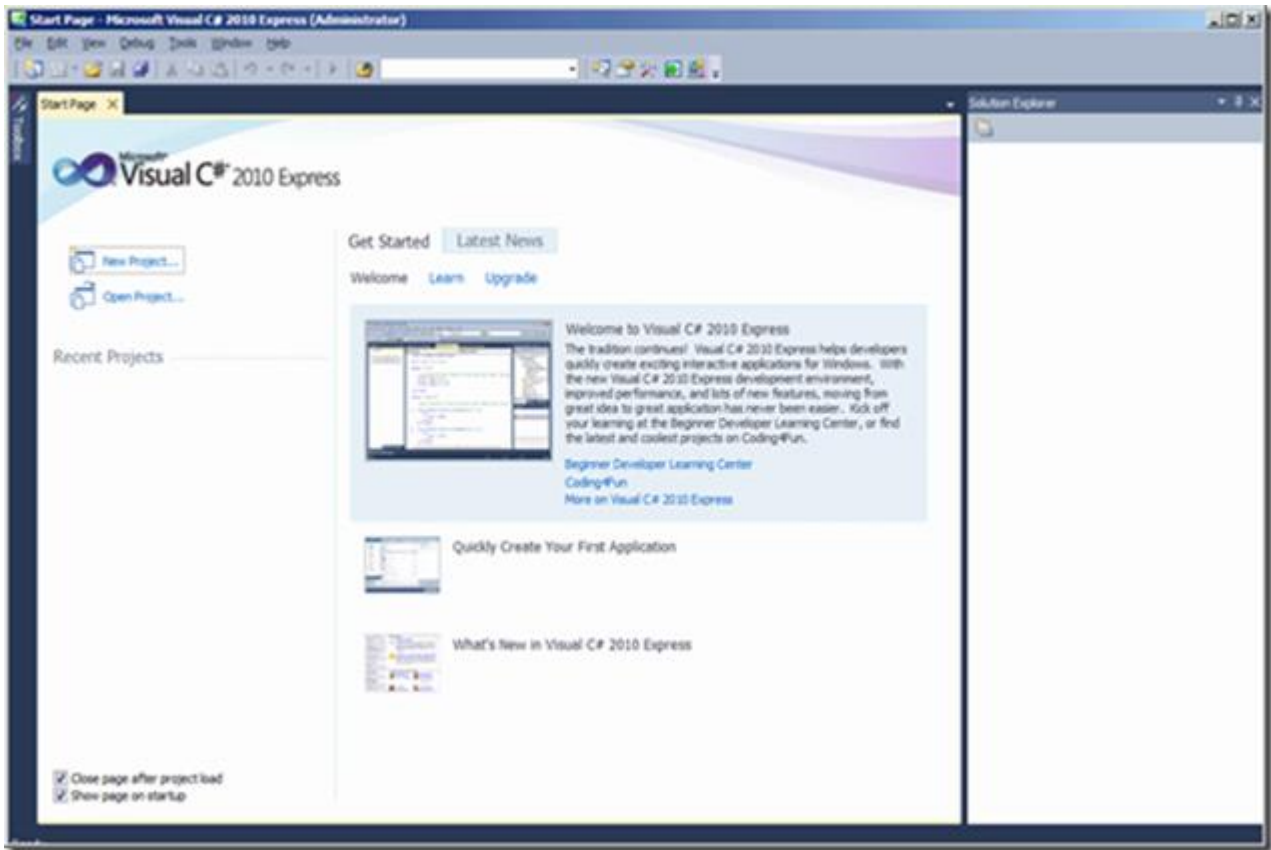
پروژه ی کامپیوتری عبارتند از گروهی فایل و شیء که دستورهای مورد نظر را برای انجام عملکرد های معین تعبیه می کند. پروژه ی **C#** متشکل از چند فایل است که دستورهای برنامه را دربردارند. یک پروژه ی ساده ی **C#** دربردارنده ی یک فایل است که تمام دستورات مورد نیاز برنامه را حمل می کند. این در حالی است که پروژه های ی پیشرفته **C#** حاوی بیش از یک فایل است.

مقدمه

برنامه های کاربردی کنسول

زبان برنامه نویسی **C#** به منظور طراحی و نوشتن برنامه های کاربردی **DOS** به کار می رود. برای طراحی چنین برنامه ای باید دستورات مورد نظر را در برنامه ی **Notepad** یا محیط برنامه نویسی مانند **visual studio** نوشت.

برای راه اندازی برنامه **Microsoft Visual C# 2010 Express**، دستورالعمل زیر را دنبال کنید : **Start -> (All) Programs -> Microsoft Visual Studio 2010 Express -> Microsoft Visual C# 2010 Express**



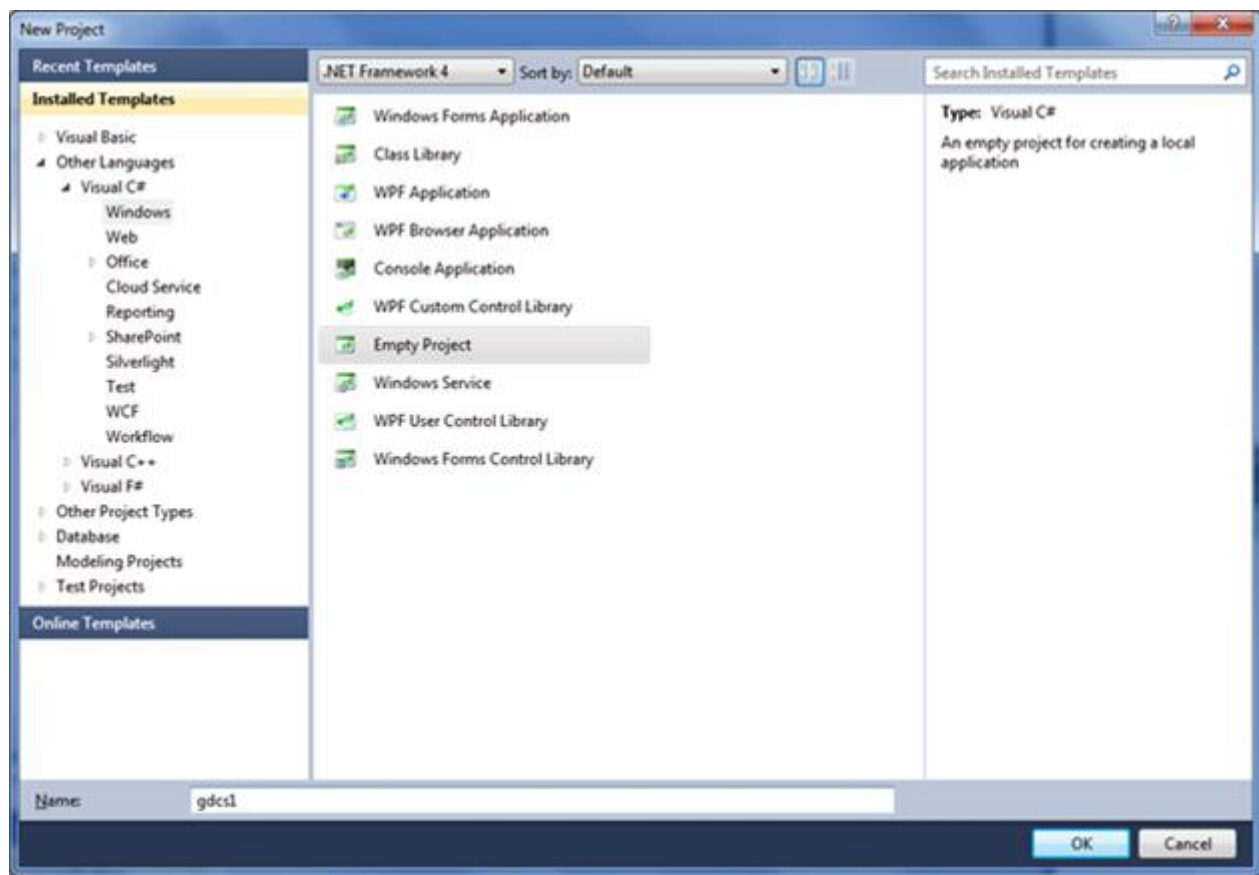
راه اندازی پروژه

برنامه ی **Microsoft Visual C# 2010 Express** را اجرا کنید.

به منظور ایجاد برنامه ی کاربردی جدید (**app**)، در فهرست گزینه ی اصلی (**main menu**) روی گزینه ی **File - New Project...** کلیک کنید.

در لیست وسطی نیز گزینه ی **Console Application** را کلیک کنید.

اسم مورد نظر را به **gdcs1** تغییر دهید.



روی گزینه ی **ok** کلیک کنید.

ایجاد فایل برای کد مورد نظر

هدف از نوشتن برنامه، دادن دستور (اجرای عملکرد خاص) به کامپیوتر است. البته، این برنامه را با زبان عادی (مثلاً انگلیسی) و جملات معنی دار می نویسیم. به عبارت دیگر، یک دستور معمولی از متن ساده استفاده می کند (که شامل حروف الفبا، ارقام و علامت های ناخوانا می باشد).

می توان این دستورات را با کمک برنامه های ویرایشگر متن (**text editor**) از قبیل **Notepad**، **Microsoft word**، **wordpad** و **wordperfect** نیز نوشت. در برنامه نویسی، قوانین و دستور هایی وجود دارد که پیروی از آن ها الزامی است. به دستورات مزبور کد هم گفته می شود.

در برنامه های **Microsoft Visual C# Express** و **Microsoft Visual Studio Professional**، ویرایشگر متنی وجود دارد که از آن با نام **code editor** یاد می شود. چنانچه، پروژه ی خود را با فرمت **console application** بنویسید، فایل پیش فرضی به وجود می آید که می توان آن را تنظیم (**customize**) کرد. اما اگر برنامه ی خود را به عنوان پروژه ای تهی آغاز کنید، لازم است خود یک فایل به آن اضافه کنید. برای این منظور دستورات زیر را انجام دهید.

Main menu را باز کرده و روی گزینه های **Project -> Add New Item...** کلیک کنید.

حال، در قسمت **solution explorer**، روی اسم پروژه ی مورد نظر راست کلیک کرده و مکان نمای موس را روی گزینه ی **Add** قرار دهید، سپس **New Item** را انتخاب کنید.

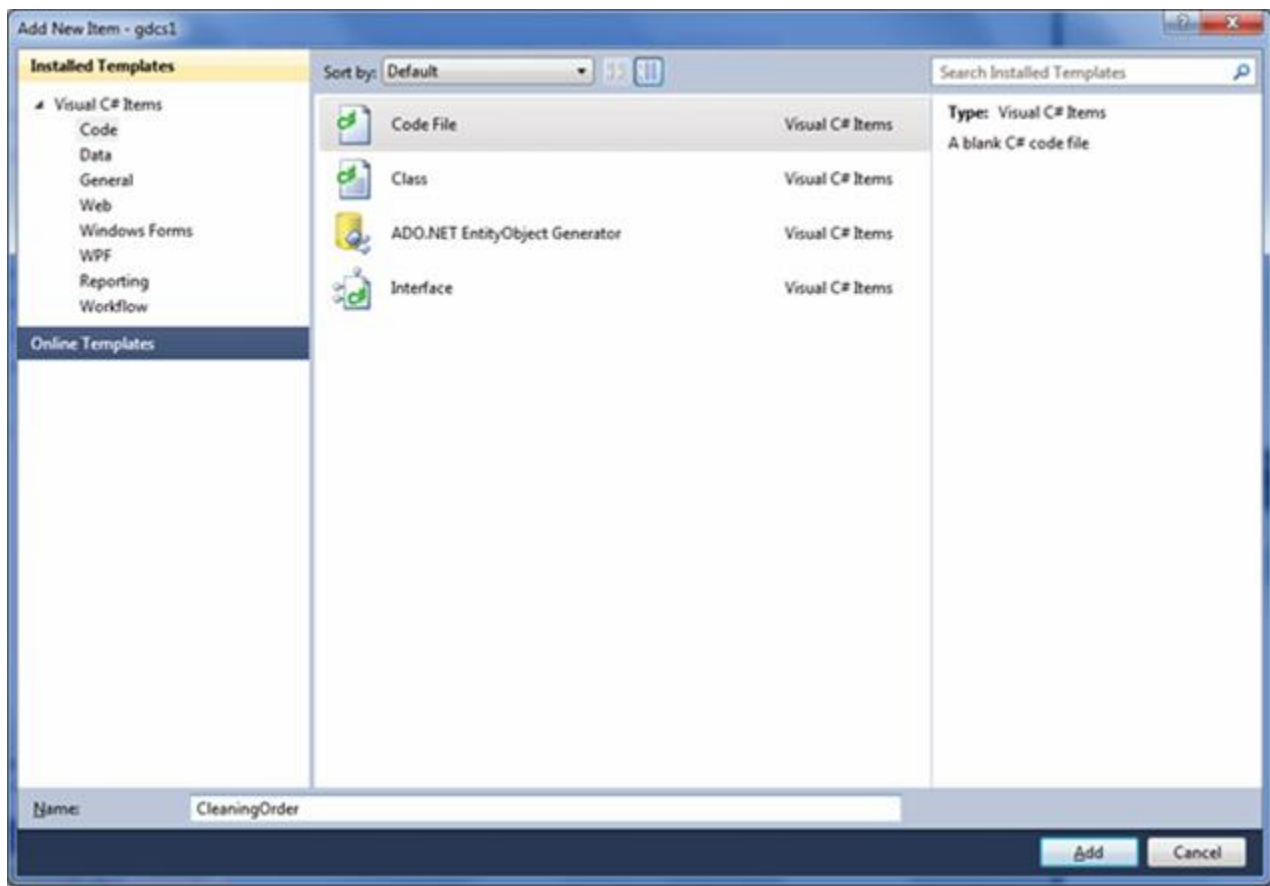
هریک از دستورهای مذکور، پنجره ی **Add new item** را نشان می دهد. حال، در لیست میانی، روی گزینه ی **Code File** کلیک کنید. می توانید اسم فایل مذکور را بپذیرید یا آن را مطابق میل تغییر دهید. پس از کلیک روی گزینه ی **ok**، پنجره ی سیاهی نمایان می گردد. به همین شیوه، می توانید هر تعداد فایل که مایلید به پروژه اضافه کنید.

ایجاد فایل

به منظور افزودن فایل جدید به کد مورد نظر، به فهرست گزینه ی اصلی (**main menu**) مراجعه کرده و روی **Project -> Add New Item...** کلیک کنید.

در صورت استفاده از **visual studio**، باید آیتم های **visual C#** (در لیست سمت چپ) را بزرگ نمایی کرده و روی **code** کلیک کنید.

اسم مورد نظر را به **CleaningOrder** تغییر دهید.



اکنون، گزینه ی Add را انتخاب کنید.

کد اصلی C#

به منظور ایجاد ساده ترین نوع برنامه ی کاربردی C#، کافی است تنها یک کد پایه (basic) بنویسید. کد مذکور به این صورت است.

```

class Exercise
{
    static void Main()
    {
        System.Console.WriteLine("Hello world");
    }
}

```

کد نام برده بخش های مختلفی دارد که در مباحث بعدی به هریک از آن ها مفصل پرداخته می شود.

کلید واژه ی class در درس 5 معرفی می شود.

واژه ی **Exercise** تنها یک اسم است. درباره ی اسم ها و تمام قوانین مربوط به آن در درس بعدی بحث می کنیم.

علامت های {}, (), در درس 4 معرفی می شوند.

نحوه ی استفاده از کلیدواژه ی **static** در درس 10 بحث می شود.

معنی کلیدواژه ی **void** در درس 5 توضیح داده می شود.

Main را در درس 7 معرفی می کنیم و در درس 26 مفصل توضیح می دهیم.

نقش واژه ی **system** در درس 8 تشریح می شود.

نقش کاراکتر نقطه "." را در درس 5 تشریح می کنیم.

Console و **waterline** در درس 6 توضیح داده می شوند.

علامت (") را در درس 4 معرفی می کنیم و مفصل در درس 28 تشریح می کنیم.

این کد را تا حد امکان ساده معرفی کرده ایم. تک تک کلمات و عملگرها در این کد وظیفه ی خاصی را انجام می دهد ولی می توان آن ها را به شیوه های مختلف به کاربرد.

نوشتن کد اصلی

کد زیر را در داکيومنت خالی بنویسید.

```
class Order
{
    static void Main()
    {
        System.Console.WriteLine("Georgetown Dry Cleaning
Services");
        System.Console.ReadKey();
    }
}
```

Comments

خط یا پاراگرافی از متن که جزء کد محسوب نمی گردد. در زبان **C#** دو نوع **comment** وجود دارد. برای نشان دادن / قرار دادن **comment** در (خطی از) متن، خط مورد نظر را با علامت **//** آغاز کنید. هر چه در سمت راست این علامت قرار بگیرد، نادیده گرفته می شود. به مثال زیر توجه کنید.

```
// This line will be ignored. I can write in it anything I want
```

Comment مذکور در تنها یک خط قابل استفاده است. می توان **comment** را با علامت **/*** نیز آغاز کرد. این نوع **comment** به علامت ***/** ختم می شود. هر چه بین این دو علامت **/*** و ***/** قرار گیرد، خوانده نمی شود. با این روش می توان، **comment** را در چند خط (بیش از یک خط) به کار برد. به منظور قرار دادن (افزودن) **comment** در خط، ابتدا روی خط مورد نظر کلیک کنید. حال، در (قسمت) **Standard toolbar**، گزینه ی **Comment Out the Selected Lines** را انتخاب کنید. برای افزودن **comment** به خط های مجاور، روی متن خط های نام برده راست کلیک کرده، سپس روی گزینه ی **Comment Out the Selected Lines** کلیک کنید. به منظور حذف **comment** ها، ابتدا روی متن آن کلیک کنید. سپس، گزینه ی **Uncomment the Selected Lines** را از **Standard toolbar** انتخاب کنید.

ایجاد **comment**

به منظور ایجاد **comment**، فایل مورد نظر را به صورت زیر تغییر دهید.

```
// Application Name: Georgetown Dry Cleaning Services
class Order
{
    // This application is used to perform processing
    orders for this business
    static void Main()
    {
        /* Always greet the customers before processing
an order.
        Check the types of clothes a customer has brought.
        Find out the type of cleaning (starch, etc) the
customer wants.
        If there are special needs the customer has, make a
note. */
```

```
System.Console.WriteLine("Georgetown Dry Cleaning
```

```
Services" );  
    System.Console.ReadKey ();  
    }  
}
```

مدیریت فایل ها

مقدمه

در برنامه های **Microsoft Visual C# 2010 Express** و **Microsoft Visual Studio**، پروژه هایی که متشکل از چند فایل هستند، با ورچسب های معینی (واقع در قسمت بالای **code editor**) نمایش داده می شوند. فایل های نام برده در **main menu** ویندوز نیز قابل مشاهده است.

Solution Explorer

Solution explorer پنجره ای است که فهرستی از فایل های تشکیل دهنده ی پروژه را به نمایش می گذارد. برای دسترسی به پنجره ی مزبور

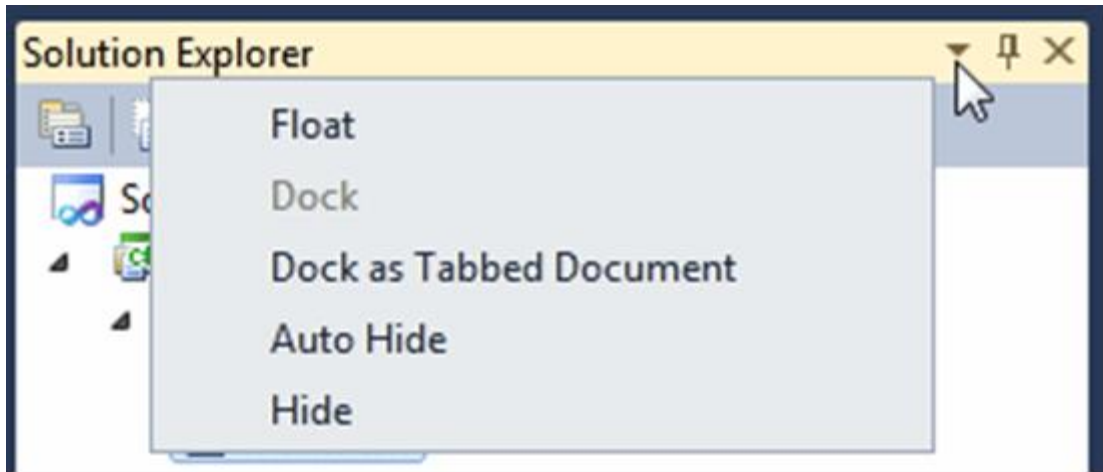
چنانچه، **Solution explorer** در صفحه ی مانیتور نمایان نیست.

به **main menu** مراجعه کرده، سپس روی گزینه ی **View -> Solution Explorer** کلیک کنید.

Standard toolbar را باز کرده، و گزینه ی **Solution Explorer** را انتخاب کنید.

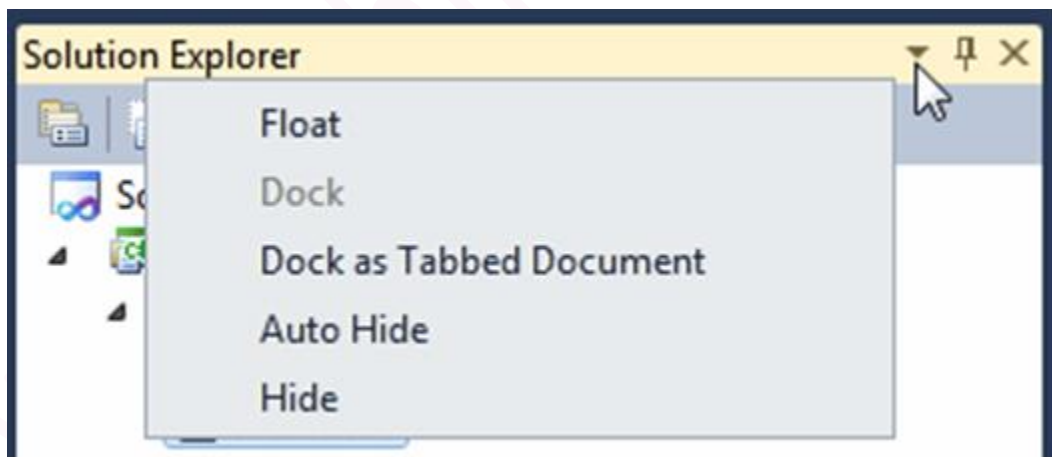
چنانچه، **Solution Explorer** در صفحه قابل رویت است، روی تب آن کلیک کنید.

این پنجره از چهار جز تشکیل می شود. درست مانند پنجره های دیگر، بخشی دارد به نام نوار تیتلر (**title bar**) که در سمت چپ آن سه گزینه وجود دارد.



با کلیک روی دکمه ی موقعیت پنجره (**window position**)، فهرستی پدیدار می گردد.

در صورتی که پنجره ی مورد نظر به کناره ی صفحه متصل باشد، گزینه ی **float** فعال می باشد. چنانچه پنجره از قبل در حالت شناور قرار دارد، گزینه ی **float** فعال نمی باشد. پس از کلیک روی گزینه ی **float** پنجره ی (متصل به کناره ی صفحه) شناور می شود. پی در پی، می توانید با گرفتن و کشیدن **titlebar** به وسط صفحه، آن را از حالت متصل (**docked**) خارج کنید.



به منظور متصل کردن پنجره به کناره ی صفحه، **titlebar** را بکشید. زیر سرتیتر (**titlebar**)، فهرستی وجود دارد که به آن نوار ابزار (**toolbar**) می گویند



گزینه ی Properties

گزینه ی **show all files**؛ تمام فایل های مخفی (**hidden**) را نمایش می دهد.

دکمه ی **refresh**، فهرست منابع و فایل ها را باز سازی (**refresh**) می کند.

دکمه ی **view code**، برای نمایش کد یک کلاس به کار می رود.

سومین بخش **solution explorer** بدنه ی اصلی آن است که در آن فایل ها، فولدرها و منابع تشکیل دهنده ی پروژه به نمایش گذاشته می شود. به منظور بزرگنمایی گره (**node**)، روی اسم آن دوبار کلیک کنید (یا روی دکمه ی آن کلیک کنید). برای کوچک کردن گره، روی آن دوبار کلیک کنید.

ریشه ی (**root**) لیست، در واقع اسم **solution** است. زیر **root**، اسم پروژه ی جاری قرار دارد. در صورتی که، **solution** دارای چند پروژه باشد، اسم هر یک از پروژه ها زیر **solution** نمایش داده می شود. درون پروژه ها : فایل ها، فولدرها و منابع وجود دارد. اولین آیتم زیر اسم پروژه **References** است. پس از گره **References** (ارجاعات)، با اسم کلاس ها (ی تشکیل دهنده ی پروژه) مواجه می شویم.

چهارمین جز **solution explorer** تب (**tab**) می باشد.

Properties Windows

پنجره ی **properties**، تمام جزئیات فایل ها و منابع سیستم عامل ویندوز که در پروژه به کار برده شده را نمایش می دهد. برای نمایش آن چنانچه، پنجره ی **Properties** نمایان نباشد، به **main menu** مراجعه کرده سپس روی **View -> Properties window** کلیک کنید.

Standard toolbar را باز کرده، گزینه ی **Properties** را انتخاب کنید.

حال، اگر پنجره ی **Properties**، در صفحه ی نمایش قابل رویت است، روی تب آن کلیک کنید.

به منظور مشاهده ی ویژگی های سیستم عامل پروژه یا فایل موردنظر (در **solution explorer**)، روی شی دلخواه کلیک کنید.

با کلیک روی **solution** پنجره ی **Properties** اسم و مسیر (مکان قرارگیری فایل) آن را نمایش می دهد. چنانچه، گزینه ی **project** را انتخاب کنید، پنجره ی **Properties** فایل های پروژه ی موردنظر را نشان می دهد

اگر روی فایل کلیک کنید، پنجره ی **Properties** اسم و مسیر فایل نام برده را به نمایش می گذارد.

پنجره ی **Properties** بسته به آیتم انتخابی شما (در **solution explorer**) قسمت های (**field**) مختلفی را نشان می دهد. می توان برخی تنظیمات را در پنجره ی **Properties** دستکاری کرد. چنانچه، قسمت (**field**) ای غیر فعال شود، بدین معنا است که دیگر نمی توان آن را اصلاح کرد.

دسترسی به فایل و باز کردن آن

در برنامه های ویرایش گر متن مثل **Notepad**، بخشی وجود دارد به نام **open file**. چنانچه سعی دارید فایل **c#** را با آن باز کنید، ابتدا لازم است که در **combo box**، نوع فایل را به **All files (*)** تغییر دهید. سپس، محل قرارگیری فایل را پیدا کرده و روی **open** کلیک کنید.

چنانچه فایل مذکور عضوی از پروژه جاری است برای باز کردن آن مراحل زیر را دنبال کنید.

به **main menu** مراجعه کرده، روی **Window** و اسم فایل موردنظر کلیک کنید.

در **code editor**، روی برچسب حاوی اسم فایل کلیک کنید.

در **solution explorer**، روی اسم فایل دو بار کلیک کنید.

در صورت استفاده از **Microsoft Visual C# 2010 Express** و **Microsoft Visual Studio**، برای باز کردن

فایل در **main menu**، روی گزینه های **click File -> Open File...** کلیک کنید یا به **Standard toolbar**

مراجعه کرده و روی **Open File** کلیک کنید.

در نتیجه ی هریک از عملیات بالا **open file** ظاهر می گردد. حال، فایل را از فولدر مرتبط پیدا کرده و روی آن کلیک کنید. اگر **visual studio** امکانات باز کردن فایل را داشته باشد آن را داخل رابط (**interface**) خود نمایش می دهد. ولی اگر نتواند آن را باز کند، در کامپیوتر به دنبال برنامه ای که قادر به باز کردن فایل مزبور است می گردد. برای مثال، اگر سعی کنید فایل ویدئویی را با **visual studio** باز کنید، برنامه ی **windows media player** به کمک شما می آید و فایل را باز می کند.