



برنامه نویسی C#

با رویکرد علمی و کاربردی

مؤلف:

دکتر محمدعلی ترکمانی

سرشناسه	: ترکمانی، محمدعلی، ۱۳۵۴ -
عنوان و نام پدیدآور	: برنامه نویسی C# با رویکرد علمی و کاربردی/ مولف محمدعلی ترکمانی.
مشخصات نشر	: مشهد: ارسطو، ۱۳۹۶.
مشخصات ظاهری	: ۳۴۶ ص.؛ مصور، جدول، نمودار.
شابک	: 978 - 600 - 432 - 185 - 3
وضعیت فهرست نویسی	: فیبا
موضوع	: سی شارپ (زبان برنامه نویسی کامپیوتر)
موضوع	: C # (Computer program language)
رده بندی کنگره	: QAV۶/۷۳ / ۴ت۹۵س / ۱۳۹۶
رده بندی دیویی	: ۰۰۵/۱۳۳
شماره کتابشناسی ملی	: ۴۸۴۰۵۳۳

نام کتاب: برنامه نویسی به زبان C#

مؤلف: دکتر محمدعلی ترکمانی

ناشر: ارسطو (با همکاری سامانه اطلاع رسانی چاپ و نشر ایران)

صفحه آرای، تنظیم و طرح جلد: علی بیات

تیراژ: ۱۰۰۰ جلد

نوبت چاپ: دوم - ۱۳۹۹

تعداد صفحات: ۳۰۷ صفحه

چاپ: مدیران

قیمت: ۴۰۰۰۰ تومان

شابک: 978-600-432-185-3

تلفن های مرکز پخش: ۰۹۱۷۷۱۶۴۹۴۰ - ۵۰۹۶۱۴۶ - ۰۵۱۱

وب سایت: www.chaponashr.ir/Torkamani

این اثر مشمول قانون حمایت از مولفان و مصنفان و هنرمندان است. هر کس تمام یا قسمتی از این اثر را بدون اجازه مولف نشر یا پخش یا عرضه کند، مورد پیگرد قانونی قرار خواهد گرفت.

فهرست مطالب

_Toc50066342

فصل اول: آشنایی با ویژوال استودیو ۱۷

۱-۱-سی شارپ چیست؟ ۱۷

۱-۲- Visual Studio ۱۷

۱-۳- ساختار .NET Framework ۱۸

۱-۴- محیط کاری ویژوال استودیو ۱۹

۱-۴-۱- منو بار (Menu Bar) ۱۹

۱-۴-۲- The Toolbars ۲۱

۱-۴-۳- صفحه آغازین (Start Page) ۲۱

۱-۵- ایجاد پروژه جدید ۲۱

۱-۶- تغییر ظاهر ویژوال استودیو ۲۳

۱-۷- سئوالات تشریحی ۳۱

۱-۸- سئوالات چهارگزینه‌ای ۳۱

پاسخنامه ۳۲

فصل دوم: نوشتن و اجرای برنامه‌ها ۳۳

۲-۱- ایجاد پروژه جدید ۳۳

۲-۲- آشنایی با پسوند فایل‌ها ۳۴

۲-۳- آشنایی با پنجره Solution Explorer ۳۶

۲-۴- نوشتن برنامه ۳۸

۲-۵- اجرای برنامه ۴۰

- ۴۱ ۲-۶- ساختار برنامه‌های C#
- ۴۱ ۲-۷- نوشتن توضیحات یا Comment برای کدها
- ۴۲ ۲-۸- ترجمه و اجرای برنامه بدون استفاده از Visual studio
- ۴۲ ۲-۹- استفاده از IntelliSense
- ۴۴ ۲-۱۰- رفع خطاها
- ۴۷ ۲-۱۱- سئوالات تشریحی
- ۴۸ ۲-۱۲- سئوالات چهارگزینه‌ای
- ۴۸ پاسخنامه

فصل سوم: دستورات کنسول و متغیرها ۴۹

- ۴۹ ۳-۱- دستورات محیط Console
- ۴۹ ۳-۱-۱- دستور Console.WriteLine
- ۴۹ ۳-۱-۲- دستور Console.WriteLine
- ۵۰ ۳-۱-۳- کنترل عمل چاپ در دستور Write و WriteLine
- ۵۱ ۳-۱-۴- استفاده از @ برای نادیده گرفتن کاراکترهای کنترلی
- ۵۲ ۳-۲- تعریف استفاده از متغیرها
- ۵۳ ۳-۲-۱- متغیر رشته‌ای
- ۵۴ ۳-۲- دستور Console.ReadKey
- ۵۵ ۳-۳-۱- دستور readkey
- ۵۶ ۳-۴- دستور read
- ۵۷ ۳-۵- دستور ReadLine
- ۵۷ ۳-۵-۱- دریافت داده int با متد ReadLine
- ۵۹ ۳-۶- قالب بندی خروجی کنسول با استفاده از Placeholder

- ۶۰ ۳-۷- قالب بندی داده‌های عددی
- ۶۲ ۳-۸- کار با اعداد مبنای ۱۶
- ۶۲ ۳-۹- تعریف رنگ زمینه و رنگ قلم
- ۶۳ ۳-۱۰- سئوالات تشریحی
- ۶۳ ۳-۱۱- سئوالات چهارگزینه‌ای
- ۶۴ پاسخنامه

فصل چهارم: نوع‌های داده اولیه ۶۵

- ۶۵ ۴-۱- مقدمه
- ۶۵ ۴-۲- نوع داده رشته‌ای
- ۶۶ ۴-۳- نوع‌های داده عددی
- ۶۸ ۴-۴- نوع داده کاراکتر
- ۶۸ ۴-۵- Boolean نوع داده
- ۶۹ ۴-۶- object نوع داده
- ۶۹ ۴-۷- کلمه کلیدی var
- ۷۰ ۴-۸- کلمه کلیدی const
- ۷۱ ۴-۱۰- سئوالات تشریحی
- ۷۱ ۴-۱۱- سئوالات چهارگزینه‌ای
- ۷۲ پاسخنامه

فصل پنجم: عملگرهای زبان C# ۷۳

- ۷۳ ۵-۱- مقدمه
- ۷۳ ۵-۲- عملگرهای محاسباتی
- ۷۴ ۵-۳- عملگرهای رابطه‌ای

۷۵ عملگرهای منطقی
۷۶ عملگرهای بیتی
۷۹ عملگرهای انتساب
۸۰ اولویت عملگرها
۸۲ سوالات تشریحی
۸۲ سوالات چهارگزینه‌ای
۸۲ پاسخنامه

فصل ششم: دستورات کنترلی و حلقه‌ها ۸۳

۸۳ مقدمه
۸۳ دستور if
۸۵ دستور if else
۸۸ دستور if - else if
۹۲ دستور if فوری یا ؟
۹۲ دستور switch
۹۶ دستورات کنترلی و حلقه‌ها
۹۶ دستور for
۹۹ حلقه‌های تو در تو
۱۰۱ دستور break
۱۰۱ دستور continue
۱۰۱ چند مثال دیگر از دستور for
۱۰۲ دستور while
۱۰۴ حلقه‌های نامحدود
۱۰۶ حلقه do-while

۱۰۶..... ۶-۸-سوالات تشریحی

۱۰۷..... ۶-۹-سوالات چهارگزینه‌ای

۱۰۸..... پاسخنامه

فصل هفتم: آرایه‌ها و دستور foreach ۱۰۹

۱۰۹..... ۷-۱-تعریف آرایه

۱۰۹..... ۷-۲-دسترسی به خانه‌های آرایه

۱۱۱..... ۷-۳-خاصیت Length

۱۱۲..... ۷-۴-آرایه‌های چند بعدی

۱۱۳..... ۷-۵-آرایه‌های دنداندار یا نامنظم

۱۱۵..... ۷-۶-دستور foreach

۱۱۶..... ۷-۷-ایجاد آرایه‌ای از کلاس‌ها

۱۱۷..... ۷-۸-سوالات تشریحی

۱۱۸..... ۷-۹-سوالات چهارگزینه‌ای

۱۱۸..... پاسخنامه

فصل هشتم: متدها ۱۱۹

۱۱۹..... ۸-۱-آشنایی با متدها در C#

۱۱۹..... ۸-۲-شیوه تعریف متدها

۱۲۲..... ۸-۳-نحوه استفاده پارامترها در متدها

۱۲۳..... ۸-۴-مقادیر پیش فرض برای پارامترها

۱۲۳..... ۸-۵-مقدار دهی پارامترها با استفاده از نام

۱۲۴..... ۸-۶-مقادیر بازگشتی از متدها

۱۲۵..... ۸-۷-کلمات کلیدی out و ref

- ۱۲۷..... ۸-۸-آرایه به عنوان ورودی و خروجی متد
- ۱۲۸..... ۸-۹-کلمه کلیدی params
- ۱۲۹..... ۸-۱۰-سربارگذاری متدها
- ۱۳۱..... ۸-۱۱-متدهای بازگشتی
- ۱۳۲..... ۸-۱۲-سئوالات تشریحی
- ۱۳۳..... ۸-۱۳-سئوالات چهارگزینه‌ای
- ۱۳۴..... پاسخنامه

فصل نهم: متدها و کلاس‌های آماده ۱۳۵

- ۱۳۵..... ۹-۱-مقدمه
- ۱۳۵..... ۹-۲-کلاس Math
- ۱۳۷..... ۹-۲-۱-تابع Round
- ۱۳۷..... ۹-۲-۲-تابع Truncate
- ۱۳۸..... ۹-۳-کلاس String
- ۱۳۹..... ۹-۳-۱-متد Compare
- ۱۴۰..... ۹-۴-کلاس Array
- ۱۴۲..... ۹-۵-تبدیل تاریخ میلادی به شمسی
- ۱۴۳..... ۹-۶-کرونومتر
- ۱۴۳..... ۹-۷-تولید اعداد تصادفی
- ۱۴۵..... ۹-۸-سئوالات تشریحی
- ۱۴۶..... ۹-۹-سئوالات چهارگزینه‌ای
- ۱۴۶..... پاسخنامه

فصل دهم: برنامه‌نویسی شیء‌گرا ۱۴۷

- ۱۰-۱-تعریف شیء گرای و برنامه نویسی شیء گرا ۱۴۷
- ۱۰-۲-مفاهیم اساسی در برنامه نویسی شیء گرا ۱۴۸
- ۱۰-۳-آشنایی با کلاس، اشیاء و فضاهای نام ۱۵۱
- ۱۰-۳-۱- مفهوم کلاس ۱۵۱
- ۱۰-۳-۲- مفهوم شیء ۱۵۲
- ۱۰-۳-۳- روش تعریف کلاس و ساخت شیء ۱۵۲
- ۱۰-۴-آشنایی با فیلدها و متدها ۱۶۱
- ۱۰-۴-۱- Field ها ۱۶۲
- ۱۰-۴-۲- تعریف متد برای کلاس ۱۶۳
- ۱۰-۵- پیاده سازی کلاس Calculator ۱۶۴
- ۱۰-۶- کنترل عملیات انتساب و مقداردهی فیلدها ۱۶۵
- ۱۰-۶-۱- متدهای set و get ۱۶۶
- ۱۰-۶-۲- استفاده از Property ها ۱۶۸
- ۱۰-۶-۳- Automatic Properties ۱۷۲
- ۱۰-۷- سازنده ها و روش های ایجاد اشیاء ۱۷۳
- ۱۰-۷-۱- مقدار دهی اولیه شیء ۱۷۳
- ۱۰-۷-۲- سازنده ها ۱۷۴
- ۱۰-۷-۲-۱- زنجیره سازنده ها ۱۷۷
- ۱۰-۸- مخرب ها ۱۷۹
- ۱۰-۹- سئوالات تشریحی ۱۸۰
- ۱۰-۱۰- سئوالات چهارگزینه ای ۱۸۱
- ۱۸۲- پاسخنامه ۱۸۲

فصل یازدهم: وراثت و چندشکلی ۱۸۳

- ۱۱-۱-مقدمه ۱۸۳
- ۱۱-۲-تعریف وراثت و پیاده سازی آن در زبان C# ۱۸۳
- ۱۱-۳-کلمه کلیدی base ۱۸۵
- ۱۱-۴-تبدیل کلاس‌های مشتق شده به کلاس والد ۱۸۶
- ۱۱-۵-کلاس Object ۱۸۶
- ۱۱-۶-چندشکلی ۱۸۷
- ۱۱-۶-۱-متدهای virtual ۱۸۸
- ۱۱-۷-سازنده‌ها در کلاس‌های پدر و فرزند ۱۹۵
- ۱۱-۸-سطح دسترسی protected ۱۹۷
- ۱۱-۹-مخفی سازی اعضاء به وسیله کلمه کلیدی new ۱۹۸
- ۱۱-۱۰-فیلدهای readonly ۲۰۰
- ۱۱-۱۱-سئوالات تشریحی ۲۰۰
- ۱۱-۱۲-سئوالات چهارگزینه‌ای ۲۰۱
- پاسخنامه ۲۰۲

فصل دوازدهم: کلاس‌های static, partial و Extension Method ۲۰۳

- ۱۲-۱-مقدمه ۲۰۳
- ۱۲-۲-کلاس‌ها و اعضای static ۲۰۳
- ۱۲-۳-سازنده‌های static ۲۰۷
- ۱۲-۴-Extension Method ها ۲۰۸
- ۱۲-۵-کلاس‌های partial ۲۱۰
- ۱۲-۶-سئوالات تشریحی ۲۱۱

۲۱۲ ۱۲-۷-سئوالات چهارگزینه‌های

۲۱۲ پاسخنامه

فصل سیزدهم: کلاس‌های **sealed** و **abstract** ۲۱۳

۲۱۳ ۱۳-۱-مقدمه

۲۱۳ ۱۳-۲-کلاس‌ها و اعضاء **abstract**

۲۱۵ ۱۳-۳-کلاس‌ها و اعضای **sealed**

۲۱۷ ۱۳-۴-سئوالات تشریحی

۲۱۷ ۱۳-۵-سئوالات چهارگزینه‌ای

۲۱۸ پاسخنامه

فصل چهاردهم: **Value Types** و **Reference Types** ۲۱۹

۲۱۹ ۱۴-۱-مقدمه

۲۱۹ ۱۴-۲-انواع حافظه مورد استفاده در تخصیص متغیر

۲۱۹ ۱۴-۳- **Value Types**

۲۲۲ ۱۴-۴- **Reference Types**

۲۲۴ ۱۴-۵- تفاوت **Value Type** و **Reference Type** هنگام استفاده

۲۲۵ ۱۴-۶- تعریف **Value Type** ها با کمک **struct**

۲۲۸ ۱۴-۷- رشته‌ها **Reference Type** هستند یا **Value Type**

۲۲۸ ۱۴-۸- آشنایی با **null** و متغیرهای **nullable**

۲۳۰ ۱۴-۹- سئوالات تشریحی

۲۳۰ ۱۴-۱۰- سئوالات چهارگزینه‌ای

۲۳۲ پاسخنامه

فصل پانزدهم: **interface** ها ۲۳۳

۲۳۳ ۱۵-۱-مقدمه
۲۳۳ ۱۵-۲-تعریف interface ها
۲۳۵ ۱۵-۳-پیاده سازی interface ها به صریح و ضمنی
۲۳۸ ۱۵-۴-سئوالات تشریحی
۲۳۸ ۱۵-۵-سئوالات چهارگزینه ای
۲۳۹ پاسخنامه

فصل شانزدهم:تبدیل نوع ۲۴۱

۲۴۱ ۱۶-۱-مقدمه
۲۴۱ ۱۶-۲-تبدیل رشته به عدد
۲۴۱ ۱۶-۳-تبدیل عدد به رشته
۲۴۱ ۱۶-۴-انواع تبدیل نوع
۲۴۳ ۱۶-۴-۱-تبدیل کلاس ها
۲۴۳ ۱۶-۵-کلمات کلیدی checked و unchecked
۲۴۴ ۱۶-۶-استفاده از کلاس های Helper برای تبدیل نوع داده ها
۲۴۵ ۱۶-۷-کلمات کلیدی is و as
۲۴۶ ۱۶-۸- مفاهیم boxing و unboxing
۲۴۶ Boxing - ۱۶-۸-۱
۲۴۷ Unboxing - ۱۶-۸-۲
۲۴۸ ۱۶-۹-سئوالات تشریحی
۲۴۸ ۱۶-۱۰-سئوالات چهارگزینه ای
۲۵۰ پاسخنامه

فصل هفدهم:سربارگذاری عملگرها و cast ها ۲۵۱

۲۵۱	۱۷-۱-مقدمه
۲۵۱	۱۷-۲- مفهوم سربارگذاری عملگر
۲۵۵	۱۷-۳- اپراتورهای قابل سربارگذاری و غیرقابل سربارگذاری
۲۵۶	۱۷-۴-تعریف Cast های دلخواه
۲۵۸	۱۷-۵-سئوالات تشریحی
۲۵۸	۱۷-۶-سئوالات چهارگزینه‌ای
۲۵۹	پاسخنامه

فصل هجدهم: Generic ۲۶۱

۲۶۱	۱۸-۱-مقدمه
۲۶۲	۱۸-۲- Generic
۲۶۵	۱۸-۳- Constraint ها
۲۶۸	۱۸-۴-کلاس‌های Generic موجود در کتابخانه .NET
۲۶۸	۱۸-۴-۱- کلاس جنریک List
۲۷۰	۱۸-۴-۲- کلاس Dictionary
۲۷۱	۱۸-۵-سئوالات تشریحی
۲۷۲	۱۸-۶-سئوالات چهارگزینه‌ای
۲۷۳	پاسخنامه

فصل نوزدهم:استثناءها و مدیریت خطاها ۲۷۵

۲۷۵	۱۹-۱-مقدمه
۲۷۵	۱۹-۲-انواع خطاها
۲۷۵	۱۹-۳-ساختار try...catch
۲۷۷	۱۹-۳-۱- استفاده از بلوک finally

- ۲۷۸ ۱۹-۴-مدیریت خطاها به صورت عمومی
- ۲۷۹ ۱۹-۵-ایجاد استثناء با دستور throw
- ۲۸۱ ۱۹-۶-زیر کلاس ApplicationException
- ۲۸۲ ۱۹-۷-سئوالات تشریحی
- ۲۸۳ ۱۹-۸-سئوالات چهارگزینه ای
- ۲۸۴ پاسخنامه

فصل بیستم: delegate ۲۸۵

- ۲۸۵ ۲۰-۱-مقدمه
- ۲۸۵ ۲۰-۲-تعریف delegate
- ۲۸۷ ۲۰-۳-نکات delegate
- ۲۸۸ ۲۰-۴-Multicast Delegate
- ۲۸۹ ۲۰-۵-سئوالات تشریحی
- ۲۸۹ ۲۰-۶-سئوالات چهارگزینه ای
- ۲۹۰ پاسخنامه

فصل بیست و یکم: فایل‌ها و دایرکتوری‌ها ۲۹۱

- ۲۹۱ ۲۱-۱-مقدمه
- ۲۹۱ ۲۱-۲-کار با فایل‌ها
- ۲۹۲ ۲۱-۳-متد writeByte
- ۲۹۳ ۲۱-۴-متد ReadByte
- ۲۹۳ ۲۱-۵-متد close
- ۲۹۳ ۲۱-۶-متد position
- ۲۹۴ ۲۱-۷-متد Append

- ۲۹۴.....Write و Read با متدهای نوشتن و خواندن۲۱-۸
- ۲۹۶.....StreamReader و StreamWriter با استفاده از کلاس نوشتن فایل۲۱-۹
- ۲۹۶.....خواندن فایل به صورت بلاکی۲۱-۱۰
- ۲۹۷.....کار با دایرکتوری‌ها۲۱-۱۱
- ۲۹۷.....برگرداندن فایل‌های موجود در مسیر۲۱-۱۱-۱
- ۲۹۷.....برگرداندن فولدرهای موجود در مسیر۲۱-۱۱-۲
- ۲۹۷.....تعیین درایوهای یک کامپیوتر۲۱-۱۱-۳
- ۲۹۸.....تعیین نوع درایو۲۱-۱۱-۴
- ۲۹۹.....ایجاد فولدر۲۱-۱۱-۵
- ۲۹۹.....Get Current Directory متد۲۱-۱۱-۶
- ۳۰۰.....تعیین ظرفیت کل و ظرفیت خالی دیسک۲۱-۱۱-۷
- ۳۰۱.....تعیین نوع فایل سیستم۲۱-۱۱-۸
- ۳۰۱.....کار با Attribute فایل۲۱-۱۱-۹
- ۳۰۲.....سئوالات تشریحی۲۱-۱۲
- ۳۰۲.....سئوالات چهارگزینه‌ای۱۳-۲۱
- ۳۰۳.....پاسخنامه
- ۳۰۵.....منابع:

مقدمه:

زبان برنامه نویسی #C یکی از قدرتمندترین و متداول ترین زبانهای برنامه نویسی است که امروزه در پروژه های مختلف توسعه نرم افزار مورد استفاده قرار می گیرد. این کتاب با هدف آموزش کاربردی زبان برنامه نویسی #C تدوین شده است. از این کتاب می توان به عنوان مرجع دروس برنامه سازی شیء گرا، برنامه سازی پیشرفته، و سایر دروس برنامه نویسی در رشته های فناوری اطلاعات و کامپیوتر استفاده نمود. امید است این اثر مورد توجه اساتید، دانشجویان و علاقه مندان به زبانهای برنامه نویسی قرار گیرد. از خوانندگان عزیز تقاضا دارم نقطه نظرات خود را از طریق ایمیل m.a.torkamani@gmail.com با اینجانب در میان بگذارند تا انشالله در ویرایش های بعدی اشکالات یا کاستی های احتمالی کتاب مورد تجدیدنظر قرار گیرد. در پایان از زحمات آقای مهندس علی بیات به خاطر طراحی جلد کتاب و همچنین از مدیریت انتشارات ارسطو جناب آقای حسین قنبری تشکر و قدردانی نماییم.

محمدعلی ترکمانی

تابستان ۱۳۹۹

فصل اول

آشنایی با ویژوال استودیو

۱-۱- سی شارپ چیست؟

سی شارپ (C#) یک زبان برنامه نویسی سطح بالا، شیء گرا و همه منظوره است که توسط مایکروسافت ساخته شده و ترکیبی از قابلیت‌های خوب ++C و JAVA است. از C# می‌توان برای ساخت برنامه های تحت ویندوز، تحت وب، وب سرویس ها، برنامه های موبایل و بازی‌ها استفاده کرد. C# از کتابخانه کلاس NET. که شامل مجموعه بزرگی از اجزاء از قبل ساخته شده است، استفاده می‌کند. برای اجرای یک برنامه C# ابتدا باید NET Framework نصب شود.

۱-۲- Visual Studio

برای نوشتن برنامه های C#، به محیطی نیاز داریم که بتوانیم داخل آن، پروژه ها و فایل ها را مدیریت کنیم، برنامه را اجرا کنیم و به خطایابی برنامه بپردازیم. به این ابزارها IDE می‌گویند. معروفترین و بهترین ابزار برای نوشتن برنامه های C#، محیط Visual Studio است. با نصب Visual Studio، NetFramework. و زبان C# بر روی سیستم شما نصب خواهند شد.

معمولاً ابزارهای رایج در IDE ها عبارتند از :

- ویرایشگر کد یا
- ابزارهای خودکارسازی Build
- خطایاب یا همان Debugger

1 Integrated Development Environment

2 Source Code Editor

3 Build Automation Tools

Visual Studio یک IDE است که در آن می‌توان به زبان‌هایی مثل C++ و Visual Basic و #C و ... برنامه نوشت. Visual Studio در نسخه‌های مختلفی ارائه شده است. ما از نسخه Enterprise استفاده می‌کنیم.

۳-۱- ساختار .NET Framework

کتابخانه .NET مجموعه‌ای از کلاس‌های آماده از پیش نوشته شده است که در تمامی زبان‌های مبتنی بر .NET قابل استفاده هستند. شکل ۱-۱ ساختار .NET را نشان می‌دهد. در بالای شکل زبان‌هایی قرار دارند که شما می‌توانید به وسیله آنها برنامه‌های خود را مبتنی بر دات بنویسید. در مرکز این شکل کتابخانه دات نت قرار دارد. این کتابخانه مجموعه‌ای از کدهای از پیش نوشته شده است که برنامه نویسی را برای شما راحت تر می‌کند. بعد از کتابخانه، CLR قرار دارد که وظیفه تبدیل کدهای شما به زبان ماشین، اجرای برنامه‌ها و مدیریت برنامه‌های تحت دات نت را بر عهده دارد. تبدیل کدهای IL به کد ماشین، کنترل و مدیریت اجرای برنامه‌ها، مدیریت حافظه و کنترل و مدیریت امنیت در برنامه‌های دات نت از وظایف CLR هست.

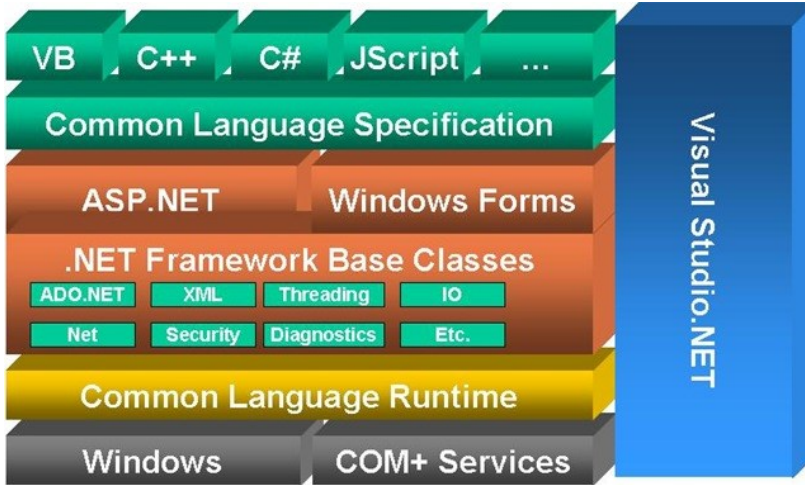
در پایین‌ترین لایه، سرویس‌های ویندوز قرار دارد که وظیفه ارتباط با این لایه، به عهده CLR و کتابخانه دات نت هست. کتابخانه دات نت مجموعه‌ای از کدهای با قابلیت استفاده مجدد است که در اختیار برنامه نویسان قرار گرفته‌اند. دومین مولفه در شکل ۱-۱، CLS است که در حقیقت استاندارد استنداردی است که با رعایت این استاندارد، کلیه زبان‌های برنامه نویسی مبتنی بر دات نت، قابلیت ارتباط با یکدیگر را خواهند داشت. برای مثال زبان‌های VB.NET و #C دو زبان مبتنی بر دات نت هستند. زمانی که شما کدی را با زبان #C نوشته‌اید، با CLS این قابلیت به شما داده می‌شود که بتوانید از کد نوشته شده در زبان #C در زبان VB.NET نیز استفاده کنید. کدهایی که با استاندارد CLS تطابق داشته باشند، بعد از کامپایل به یک کد میانی به نام IL یا MSIL ترجمه می‌شوند. کد IL شبیه به کد Assembly هست. بعد از ایجاد کد IL توسط کامپایلر، کد IL توسط CLR به کد ماشین ترجمه شده و اجرا می‌شود. مولفه دیگر CLI

1 Common Language Specification

2 Intermediate Language

۳Microsoft Intermediate Language

^۱ هست. این استاندارد یک سری خصوصیات مشترک در بین کدهای میانی ایجاد شده توسط کامپایلرها را تعریف می‌کند که برای اجرای کد IL به آنها نیاز است. برای مثال سیستم نوع داده مشترک^۲ یا سرویس های از قبیل Type Safety و اجرای مدیریت شده کدها از خصوصیات آن هستند.



شکل ۱-۱: ساختار .NET.

۴-۱- محیط کاری و ویژوال استودیو

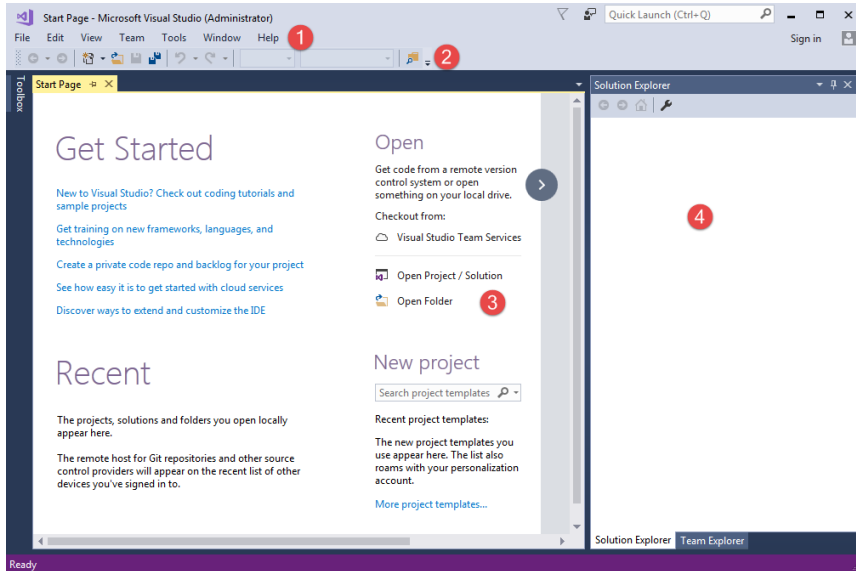
در این بخش محیط کاری و ویژوال استودیو مورد بررسی قرار می‌گیرد. برنامه ویژوال سی شارپ را اجرا کنید. محیط کار و ویژوال استودیو مطابق شکل ۱-۲ است.

۱-۴-۱- منو بار (Menu Bar)

Menu Bar (قسمت ۱ در شکل ۱-۲) 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 می‌شوند. به این نکته توجه کنید که منو دارای آیتم های مختلفی است که فقط در شرایط خاصی

1 Common Language Infrastructure
2 Common Type System

ظاهر می‌شوند. به عنوان مثال آیتم های منوی Project در صورتی نشان داده خواهند شد که پروژه فعال باشد. در جدول ۱-۱ برخی از ویژگی‌های منوها آمده است.



شکل ۱-۲ : محیط کار ویژوال استودیو

جدول ۱-۱ : برخی از ویژگی‌های منوها

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 آیمتی 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 ویژوال استودیو می باشد

۲-۴-۱- The Toolbars

Toolbar (قسمت ۲ در شکل ۱-۲) به طور معمول شامل همان دستوراتی است که در داخل منوها قرار دارند. Toolbar همانند یک میانبر عمل می‌کند. هر دکمه در Toolbar دارای آیکنی است که کاربرد آنرا نشان می‌دهد. اگر با نشانگر ماوس بر روی هرکدام از این دکمه‌ها مکت کوتاه‌های کنید، کاربرد آن به صورت یک پیام (tool tip) نشان داده می‌شود. برخی از دستورات مخفی هستند و تحت شرایط خاص ظاهر می‌شوند. همچنین می‌توانید با کلیک راست بر روی منطقه خالی از Toolbar و یا از مسیر View > Toolbars دستورات بیشتری به آن اضافه کنید. برخی از دکمه‌ها دارای فلش‌های کوچکی هستند که با کلیک بر روی آنها دیگر دستورات وابسته به آنها ظاهر می‌شوند. سمت چپ هر Toolbar به شما اجازه جابجایی آن را می‌دهد.

۳-۴-۱- صفحه آغازین (Start Page)

Start (قسمت ۳ در شکل ۱-۲) برای ایجاد یک پروژه و باز کردن آن از این قسمت استفاده می‌شود. همچنین اگر از قبل پروژه‌ای ایجاد کرده اید می‌توانید آن را در Recent Projects مشاهده و اجرا کنید. Visual Studio Community از تعداد زیادی پنجره و منو تشکیل شده است که هر کدام برای انجام کار خاصی به کار می‌روند.

۵-۱- ایجاد پروژه جدید

ابتدا از طریق منوی File > New Project یک پنجره فرم ایجاد کنید (شکل ۱-۳). پنجره‌ای به شکل ۱-۴ نمایش داده خواهد شد. همانطور که در شکل ۱-۵ نشان داده شده است گزینه Windows Forms Application و یک اسم برای پروژه انتخاب می‌کنیم و بر روی دکمه OK کلیک می‌کنیم تا شکل ۱-۶ ظاهر شود. بخش‌های فرم نشان داده شده در شکل ۱-۶ عبارت است از:

۱- صفحه طراحی (Design)

این صفحه در حکم یک ناحیه برای طراحی فرم‌های ویندوزی شما است. فرم‌های 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 ، برجسب ها و ... به فرمتان اضافه کنید .

۲- مرورگر پروژه (Solution Explorer)

شکل ۷-۱ پروژه و فایل‌های مربوط به آن را نشان می‌دهد. یک Solution برنامه ای که توسط شما ساخته شده است را نشان می‌دهد. ممکن است این برنامه یک پروژه ساده یا یک پروژه چند بخشی باشد. اگر Solution Explorer در صفحه شما نمایش داده نمی‌شود می‌توانید 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 می‌باشد و هنگام اجرای برنامه اجرا می‌شود. اگر بخواهید پروژه ای را که فعال نیست اجرا کنید، بر روی Solution Explorer کلیک راست کنید و سپس گزینه **Set StartUp Project** را انتخاب نمایید. Solution Explorer زیر یک Solution با ۲ پروژه را نشان می‌دهد. هر پروژه شامل فایلها و فولدرهای مربوط به خود است.

۳- پنجره خواص (Properties)

پنجره Properties خواص و رویدادهای مختلف هر آیتم انتخاب شده اعم از فرم ، فایل ، پروژه و کنترل را نشان می‌دهد. اگر این پنجره مخفی است می‌توانید از مسیر **View > Other Windows > Properties Window** یا کلید میانبر **F4** آنرا ظاهر کنید. 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 (شکل ۸-۱). این پنجره همچنین دارای رویدادهای مربوط به فرم یا کنترل انتخاب شده می‌باشد. یک رویداد (event) اتفاقی است که در شرایط خاصی پیش می‌آید مانند وقتی که بر روی دکمه (button) کلیک و یا متنی را در داخل جعبه متن (text box) اصلاح می‌کنیم. شکل ۸-۱ که با حرف A نشان داده شده است به شما اجازه می‌دهد که شی مورد نظرتان (دکمه، فرم ...) را که می‌خواهید 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 combo box ، دکمه های مفیدی قرار دارند (B). برخی از این دکمه ها در شرایط خاصی فعال می‌شوند. دکمه اول خاصیت اشیا را بر اساس دسته های مختلفی مرتب می‌کند. 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 ۸-۱ توضیحات کوتاهی در مورد خاصیت ها و رویداد ها 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.

۶-۱- تغییر ظاهر ویژوال استودیو

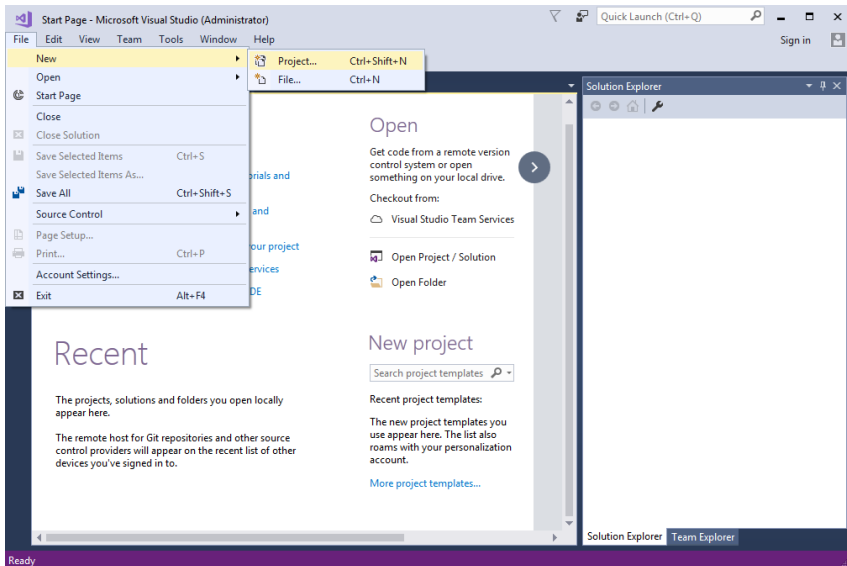
اگر موقعیت پنجره ها و یا ظاهر برنامه ویژوال استودیو را دوست نداشته باشید، می توانید به دلخواه آن را تغییر دهید. برای این کار بر روی نوار عنوان (title bar) کلیک کرده و آنرا می کشید تا پنجره به شکل ۹-۱ به حالت شناور در آید.

در حالی که هنوز بر روی پنجره کلیک کرده اید و آن را می کشید یک راهنما (فلشی با چهار جهت) ظاهر می شود و شما را در قرار دادن پنجره در محل دلخواه کمک می کند. به عنوان مثال شما می توانید پنجره را در بالاترین قسمت محیط برنامه قرار دهید. منطقه ای که پنجره قرار است در آنجا قرار بگیرد به رنگ آبی در می آید (شکل ۱۰-۱).

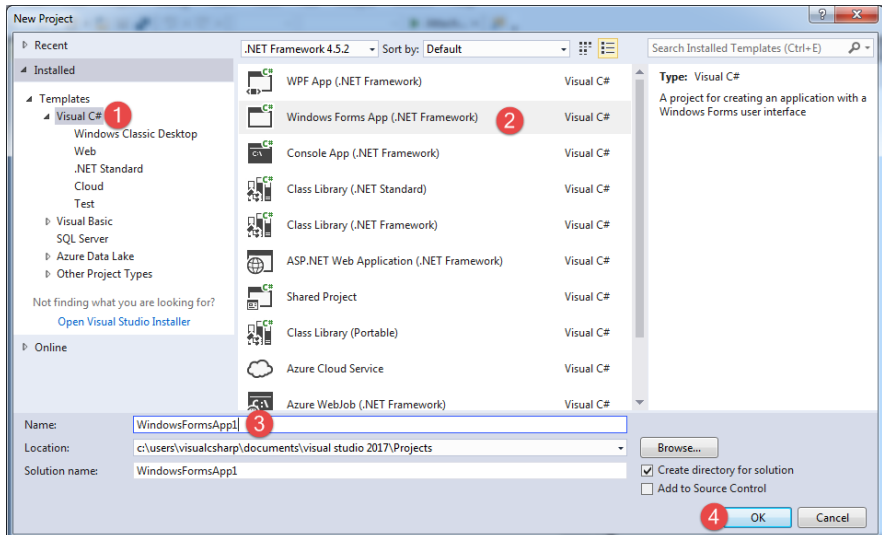
منطقه ای که پنجره قرار است در آنجا قرار بگیرد به رنگ آبی در می آید (شکل ۱۱-۱).

پنجره در قسمت بالای محیط قرار داده شده است. راهنمای صلیب شکل حاوی جعبه های مختلفی است که به شما اجازه می دهد پنجره انتخاب شده را در محل دلخواه محیط ویژوال سی شارپ قرار دهید. به عنوان مثال پنجره Properties را انتخاب کنید و آنرا به چپ ترین قسمت صلیب در پنجره نمایش داده شده نزدیک و رها کنید، مشاهده می کنید که پنجره مذکور در سمت چپ پنجره Design View قرار می گیرد (شکل ۱۲-۱).

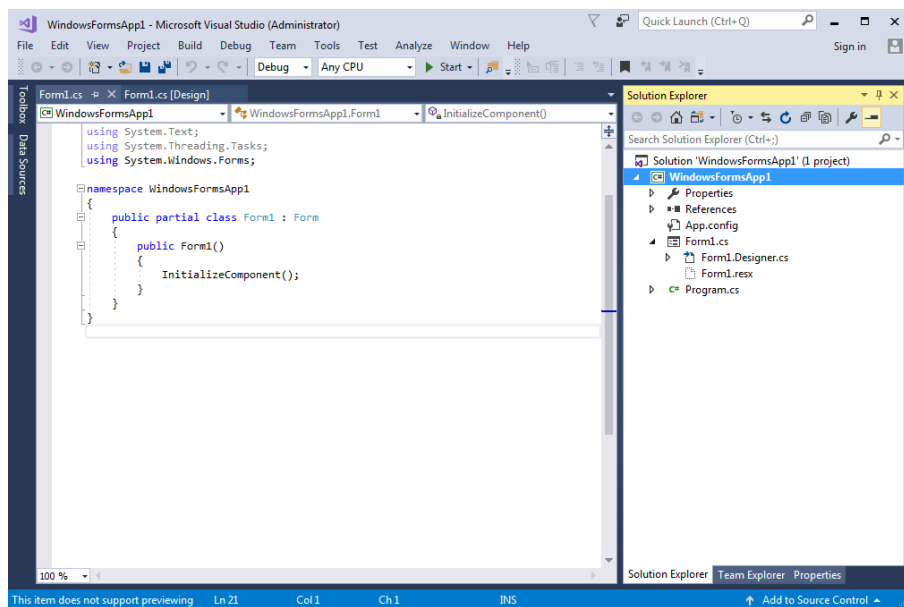
کشیدن پنجره به مرکز صلیب راهنما باعث ترکیب آن با پنجره مقصد می شود که در مثال بالا شما به توانید به عنوان یک تب به پنجره Properties دست پیدا کنید (شکل ۱۳-۱).



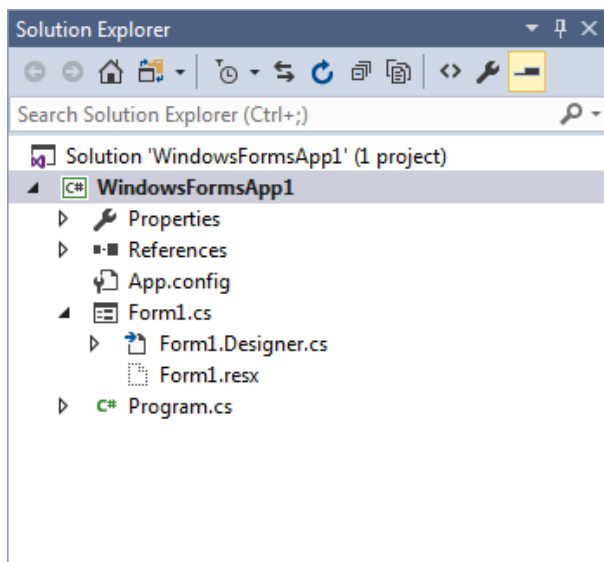
شکل ۳-۱: ایجاد پروژه جدید



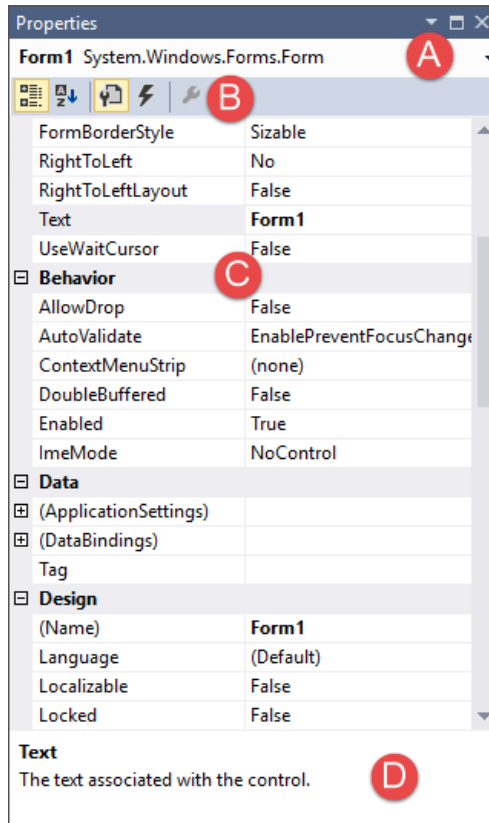
شکل ۵-۱: پنجره New Project



شکل ۶-۱ :



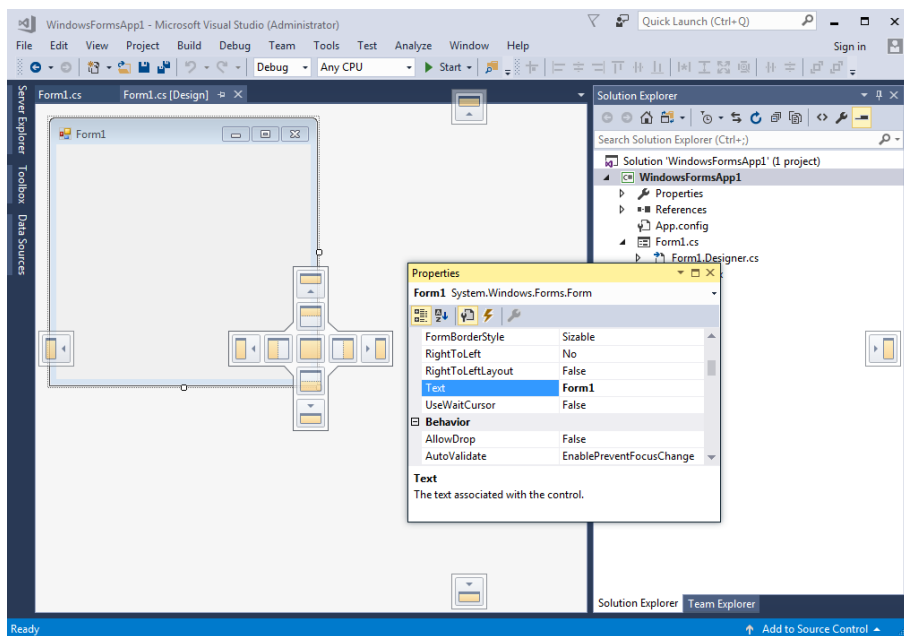
شکل ۷-۱ : پروژه و فایل‌های مربوط به آن



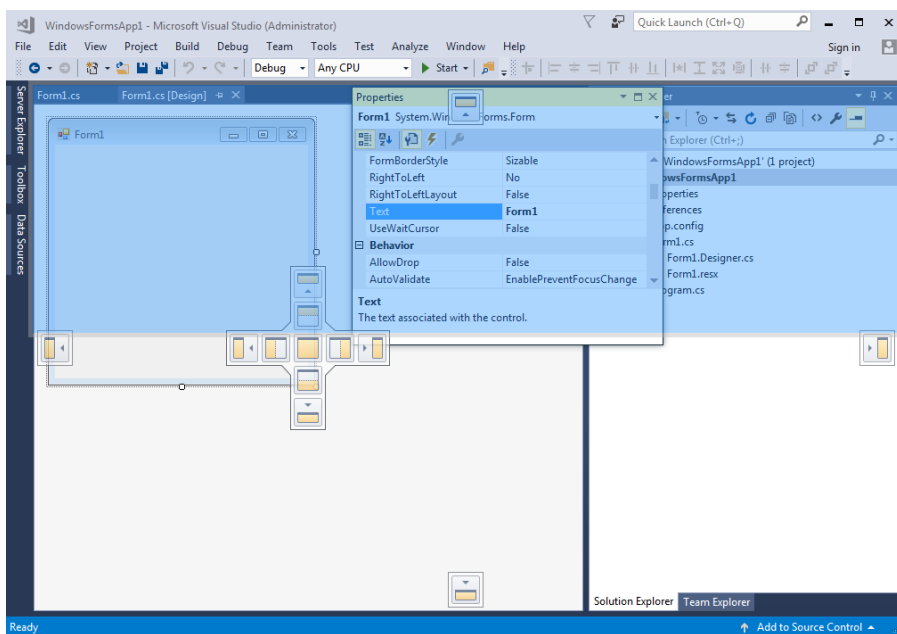
شکل ۸-۱: پنجره خواص

اگر به عنوان مثال پنجره Properties را روی پنجره Solution Explorer بکشید، یک صلیب راهنمای دیگر نشان داده می شود. با کشیدن پنجره به قسمت پایینی صلیب پنجره Properties زیر پنجره Solution Explorer قرار خواهد گرفت (شکل ۱۴-۱).

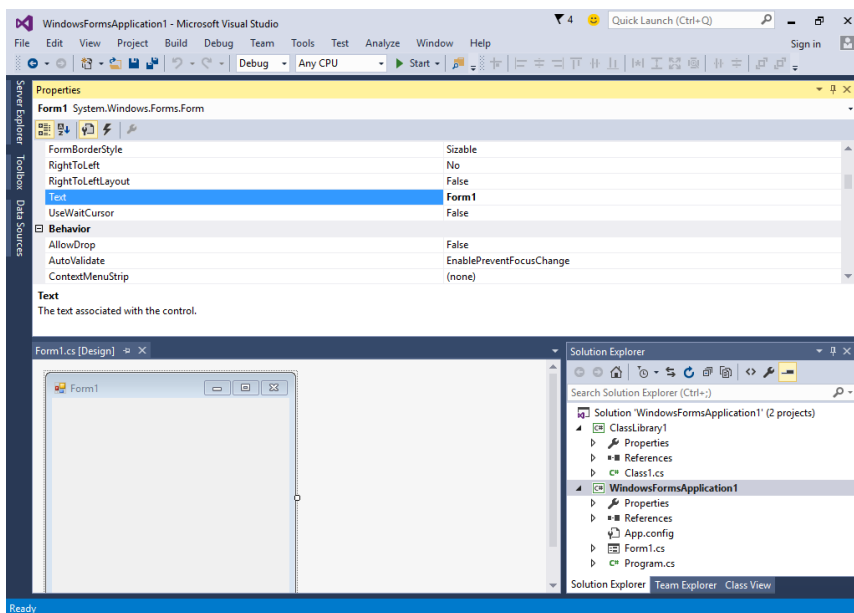
قسمتی از محیط برنامه که می خواهید پنجره در آنجا قرار بگیرد به رنگ آبی در می آید. ویژگی‌های سی شارپ همچنین دارای خصوصیتی به نام `autohide` است که به صورت اتوماتیک پنجره‌ها را مخفی می کند. هر پنجره دارای یک آیکن سنجاق مانند نزدیک دکمه `close` است (شکل ۱۵-۱). بر روی این آیکن کلیک کنید تا ویژگی `auto-hide` فعال شود. برای دسترسی به هر یک از پنجره‌ها می توان با ماوس بر روی آنها توقف یا بر روی تب های کنار محیط ویژگی‌ها سی شارپ کلیک کرد (شکل ۱۶-۱).



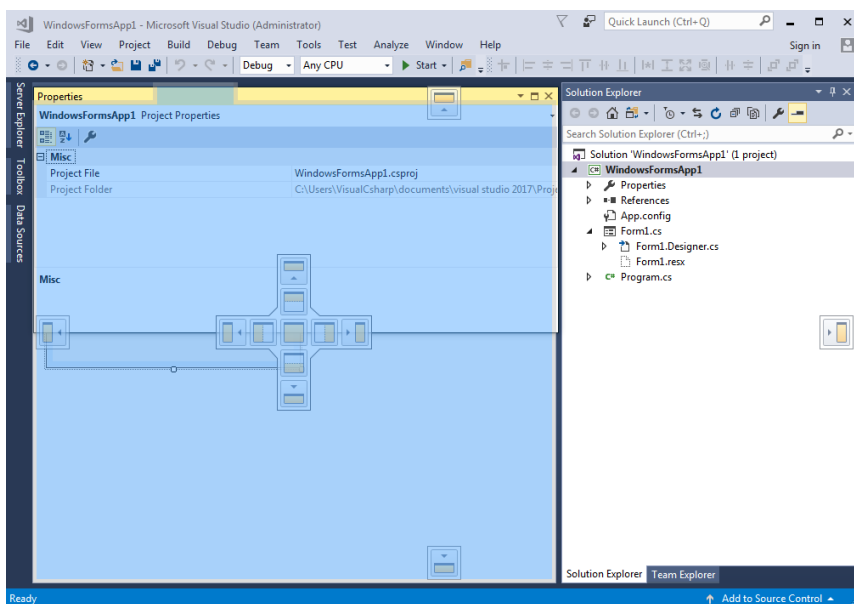
شکل ۹-۱:



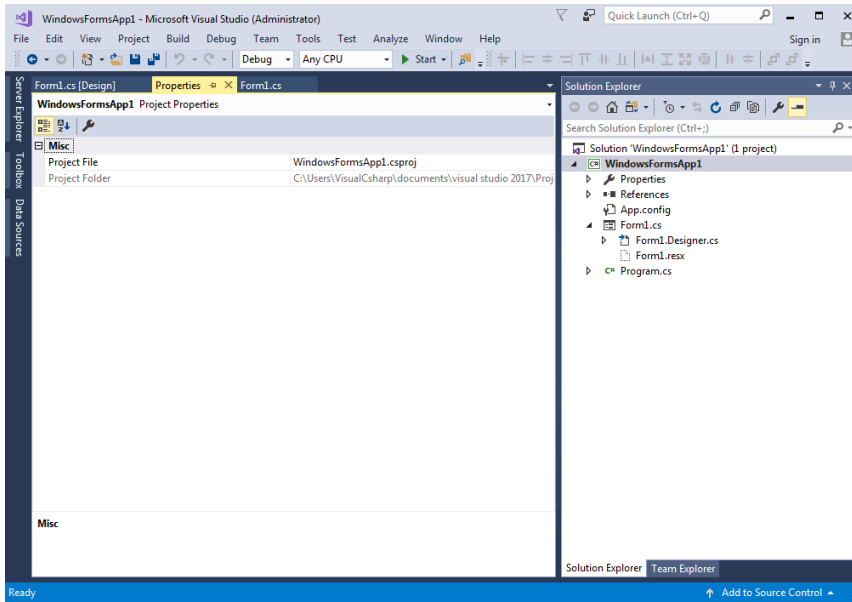
شکل ۱۰-۱:



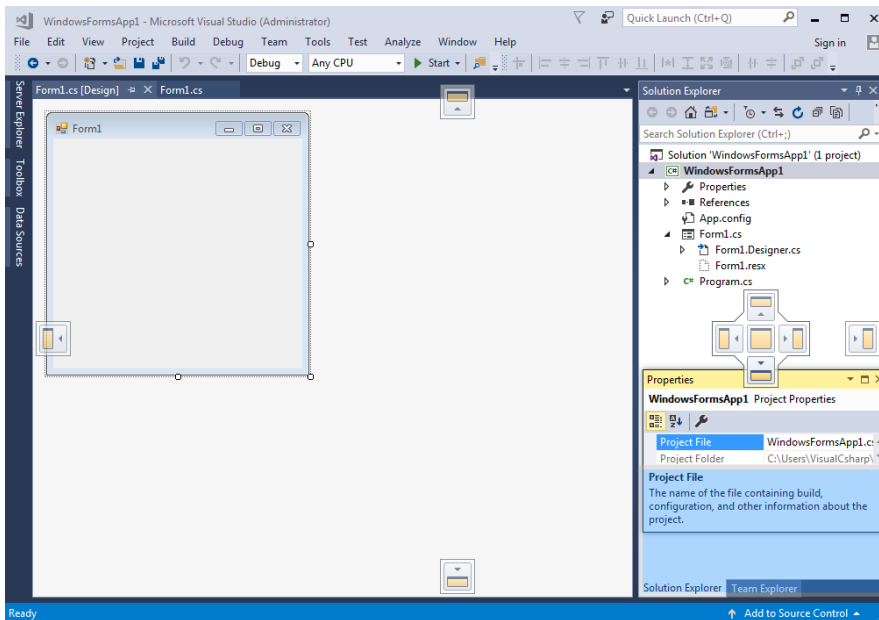
شکل ۱-۱۱ :



شکل ۱-۱۲ :

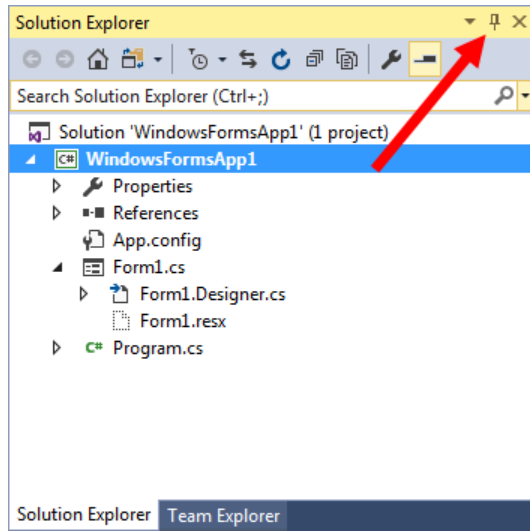


شکل ۱-۱۳ :

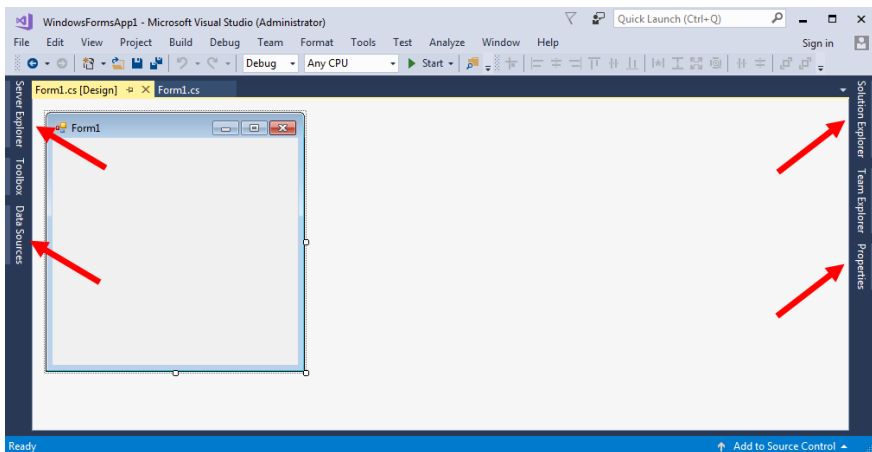


شکل ۱-۱۴ :

برای غیر فعال کردن این ویژگی در هر کدام از پنجره ها کافیتست پنجره را انتخاب کرده و دوباره بر روی آیکون مورد نظر کلیک کنید(شکل ۱۷-۱).
 به این نکته توجه کنید که اگر شکل آیکون افقی بود بدین معناست که ویژگی فعال و اگر شکل آن عمودی بود به معنای غیر فعال بود ویژگی auto-hide است.



شکل ۱۵-۱ :



شکل ۱۶-۱ :