

مرجع کامل C#

همراه با مثال‌های متنوع و کاربردی

نویسنده: Christian Nagel

مترجم: مهدی درسا



شیردانشگاهی کیان
Kian Publication

سرشناسه:

عنوان و نام پدیدآور:

مشخصات نشر:

مشخصات ظاهری:

شابک:

وضعیت فهرست‌نویسی:

یادداشت:

موضوع:

موضوع:

موضوع:

موضوع:

شناسه افزوده:

رده‌بندی کنگره:

رده‌بندی دیو:

شماره کتابشناسی ملی:

ناگل، کریستین، ۱۹۶۵ - م.

Nagel, Christian

مرجع کامل C# همراه با مثال‌های متنوع و کاربردی / نویسنده کریستین ناگل؛ مترجم مهدی مرسلی.

تهران: انتشارات دانشگاهی کیان، ۱۳۹۸.

۶۶۴ ص.

۹۷۸-۶۰۰-۳۰۷-۲۴۹-۷

فیفا

عنوان اصلی: Professional C# 7 and .net core 2.0

سی شارپ (زبان برنامه‌نویسی کامپیوتر) (C# (Computer program language)

وبگاهها -- برنامه‌های تالیفی؛ Web sites -- Authoring programs

وبگاهها -- طراحی؛ Web sites -- Design - وب -- سرورها؛ Web servers

مرسلی، مهدی، ۱۳۶۰، مترجم

QAY۶/Y۳

۰۰۵/۱۳۳

۶۰۴۵۸۵۵



نشر دانشگاهی کیان
Kian Publication

انتشارات دانشگاهی کیان

نام کتاب: مرجع کامل C# همراه با مثال‌های متنوع و کاربردی

مؤلف: Christian Nagel

مترجم: مهدی مرسلی

ویراستاران ادبی:

ناظر فنی: پیمان عمرانی

صفحه‌آرا: علی احمد اشرفی

طراح جلد: مهدی جعفری

چاپ اول: ۱۳۹۹

تیراژ: ۵۰۰

قیمت: ۹۹۵۰۰ تومان

شابک: ۹۷۸-۶۰۰-۳۰۷-۲۴۹-۷



خرید اینترنتی آسان از:

www.kianpub.com

بر اساس قانون حقوق مولفان و مصنفان، کلیه حقوق چاپ و نشر این کتاب به‌طور انحصاری به نشر دانشگاهی کیان تعلق دارد و هرگونه استفاده و برداشت از محتوای این اثر به هر شکلی اعم از چاپ، کپی، اسکن، لوح فشرده، نشر الکترونیک و اینترنتی یا به صورت هرگونه فایل رایانه‌ای، بدون مجوز رسمی ناشر ممنوع و حرام شرعی است و پیگرد قانونی دارد.



kianpublication

برای دریافت اخبار و اطلاعات مفید و شرکت در قرعه‌کشی، ما را در این شبکه‌ها دنبال کنید.

حمد و سپاس بی‌پایان، خدایی را سزااست که عالم را در منتهای کمال آفرید و آدمی را بر بلندای قله‌ی هستی گماشت و او را جلوه‌گاه جمال، مخاطب کلام و وارث قلم گردانید و «کتاب» را به عنوان بزرگ‌ترین معجزه‌ی جاودان خویش در اختیار انسان قرار داد. آیین و فرهنگ کهن ما، انس و الفتی دیرین با کتاب داشته‌اند و همواره انسان‌های وارسته، سعادت و آرامش را در کتاب جست‌وجو کرده‌اند؛ چنان‌که به تعبیر امیر مومنان علی علیه‌السلام: «کسی که با کتاب آرامش یابد، هرگز آسایش از او سلب نمی‌شود».

کتاب حافظه‌ی بشریت است و در ساحت اندیشه، هیچ وسیله‌ای همچون کتاب، واجد ژرفا و کارایی لازم نیست؛ از این‌رو ترویج و نشر کتاب، نهاده‌ی ساز‌ی فرهنگ کتاب‌خوانی، رضای حس کنجکاو و پرسشگری دانشجویان و تربیت علمی و فرهنگی نسلی شایسته و درخور، رسالتی مسلم بر دوش ارباب فرهنگ و دانش است. این رسالت، ما را نیز بر آن داشت که به عنوان عضو کوچکی از جامعه‌ی علمی و فرهنگی ایران، پای به میدان نهاده و در تحقق این هدف ارزنده و انسانی، نقشی هرچند اندک ایفا کنیم.

باعث تاسف است که در شرایط کنونی، معضلات اجتماعی و مشکلات روزمره‌ی زندگی، اشتیاق مطالعه را کاسته و با توجه به رنج‌های مجازی و شبکه‌های اجتماعی، مطالب کوتاه، سطحی و کم‌محتوا جایگزین کتاب‌های عمیق، مفید و اندیشه‌ساز گردیده و در این میان، مشکلات نشر، همچون هزینه‌های رو به افزایش کاغذ و چاپ و به تبع آن، عدم آشنایی عمیق با کتاب، مزید بر علت شده است. اگرچه این تصور که با پدید آمدن وسایل نوظهور ارتباط جمعی، کتاب، به آن خواهد رفت، تصویری خلاف واقع است، اما از این نکته نیز نباید غافل بود که امتیاز ابزارهای جدید در زمینه‌ی محاسبات، سهولت کاربری آنهاست؛ ولی به هر حال، این ابزارها هرگز از جهت عمق، تحلیل و سازندگی فکر و ذهن، جای کتاب را نخواهند گرفت.

انتشارات دانشگاهی کیان با بیش از یک دهه سابقه فعالیت در تولید و نشر کتاب‌های دانشگاهی و نیز فنی و مهندسی می‌کوشد رسالت‌های خود را در زمینه لطف پروردگار و حمایت‌های مخاطبان خود، به بهترین نحو ممکن به انجام رساند. در این راستا، تلاش بر آن است تا آنچه در این نشر به چاپ می‌رسد، حایز معیارهای استاندارد کیفی کتاب باشد و بر همین اساس است که کتاب‌ها در مسیر تولید، با حوصله و دقت تحت نظارت همه‌جانبه‌ی محتوایی-معنایی شامل چندین مرحله‌ی بازبینی علمی و نیز ادبی قرار می‌گیرند تا در نهایت متنی روان و ساده براساس اصول آموزشی تهیه و تدوین شود. همچنین برای همه آثار به لحاظ بصری و زیبایی‌شناختی از منظر صفحه‌آرایی و طراحی جلد، سطح کیفی مناسبی در نظر گرفته شده است.

همه تلاش انتشارات دانشگاهی کیان بر این بوده است تا همگام با خوانست مخاطبان خود حرکت و کتاب‌ها را با بالاترین کیفیت منتشر کند، اما به حتم، این اثر خالی از اشکال نیست و از تمامی مخاطبان فهیم آن و سایر آثار نشر خواهیمند است نقدها و نظرهای ارزشمند و سازنده خود را جهت بهبود در کتاب‌های آتی مطرح فرمایند.

نشر دانشگاهی کیان

www.kianpub.com

info@kianpub.com

.NET. پس از سال‌های طولانی حرکت جدیدی انجام داد و اکنون .NET Framework. یک خواهر کوچک‌تر به نام .NET Core دارد. .NET Framework. متن بسته و تنها روی Windows در دسترس بود؛ ولی .NET Core. متن باز و روی Linux در دسترس است و از الگوهای پیشرفته استفاده می‌کند و می‌توان بهبودهای زیادی در اکوسیستم .NET. مشاهده کرد.

با این تغییرات #C در میان ده زبان برنامه‌نویسی محبوب تمام برنامه‌نویسان، رتبه چهارم و .NET Core. رتبه سوم محبوب‌ترین چارچوب را به‌دست آورده است که می‌توان نتایج آن را در نشانی <https://insights.stackoverflow.com/survey/2017> مشاهده کرد.

با استفاده از #C و .NET Core. ASP.NET می‌توان برنامه‌های وب و سرویس‌ها را روی Windows، Linux و Mac اجرا کرد. همچنین می‌توان Windows Runtime را برای ایجاد برنامه‌های بومی Windows به‌کار برد که #C، XAML و .NET Core. از آنها استفاده می‌کنند. به‌علاوه می‌توان با Xamarin از #C و XAML برای ایجاد برنامه‌های فابل اجرا روی اندروید و iOS استفاده نمود. با کمک استاندارد .NET. می‌توان کتابخانه‌هایی ایجاد کرد و بین .NET Core. ASP.NET برنامه‌های Windows و Xamarin به‌اشتراک گذاشت. همین‌طور می‌توان برنامه‌های سنتی مبتنی بر فریم‌ورک Windows و برنامه‌های WPF ایجاد نمود که در این کتاب به بیشتر این مطالب خواهیم پرداخت.

مثال‌های این کتاب با استفاده از Visual Studio روی سیستم Windows ساخته شده‌اند. همچنین بیشتر مثال‌ها روی Linux تست شده و روی Mac، Linux و Windows اجرا شده‌اند. به جز برنامه‌های Windows می‌توان از Visual Studio Code یا Visual Studio برای Mac به‌عنوان محیط توسعه استفاده نمود.

دنیای .NET Core.

.NET. تاریخچه‌ای طولانی دارد؛ ولی .NET Core. بسیار جوان است. .NET Core 2.0. تعداد زیادی API دارد که از .NET Framework. گرفته شده‌اند تا جابه‌جا کردن برنامه‌های .NET Framework. به دنیای .NET Core. را آسان‌تر کنند.

برخی از ویژگی‌های .NET Core. عبارت‌اند از:

- .NET Core. متن باز است.
- .NET Core. از الگوهای مدرن استفاده می‌کند.
- .NET Core. از توسعه روی چند پلتفرم پشتیبانی می‌کند.
- .NET Core. ASP.NET می‌تواند روی Windows و Linux اجرا شود.

اگر با .NET Core. کار کنید، متوجه می‌شوید که این فناوری بزرگ‌ترین تغییر در .NET. را از نسخه نخست آن تاکنون به‌وجود آورده است. .NET Core. آغازی دوباره است. از این پس می‌توانیم سفر خود را برای یک توسعه سریع ادامه دهیم.

دنیای #C

#C در سال 2002 به عنوان زبانی برای توسعه .NET Framework. منتشر شد. #C با ایده‌هایی از ++C، جاوا و پاسکال طراحی شده است. آندرس هیلسبرگ از شرکت Borland به مایکروسافت آمده بود و تجربه توسعه دلفی را به همراه خود داشت. هیلسبرگ پیش از #C در مایکروسافت روی نسخه مایکروسافت جاوا کار کرد که ++J نامیده شد.

نکته

امروز هیلسبرگ به TypeScript رفته است و مدس تورگرسن¹ رهبر پروژه C# است. بهبودهای C# در نشانی <https://github.com/dotnet/csharp-lang> به اشتراک گذاشته شده‌اند و می‌توانید نشست‌ها و پیشنهادهای مربوط به C# را در آنجا دنبال کنید. همچنین می‌توانید پیشنهادهای خود برای بهبود C# را نیز ثبت کنید.

C# نه تنها یک زبان برنامه‌نویسی شیء‌گرای همه‌منظوره بلکه یک زبان برنامه‌نویسی مبتنی بر مولفه است که از خصوصیات، رویدادها، صفات (حاشیه‌نویسی‌ها) و ساخت اسمبلی‌ها (کدهای دودویی حاوی فراداده‌ها) نیز پشتیبانی می‌کند.

با گذشت زمان C# با genericها، کوثری‌های ادغام‌شده در زبان (LINQ)، عبارات لمبدا، ویژگی‌های پویا و برنامه‌ریزی ناهمگام آسان‌تر بهبود یافت. از آنجاکه C# ویژگی‌های فراوانی ارائه می‌کند، یادگیری آن چندان آسان نیست. چیزی بیش از یک زبان برنامه‌نویسی شیء‌گرا یا یک زبان برنامه‌نویسی مبتنی بر مولفه است. C# می‌آید بهیچ‌یک از برنامه‌نویسی تابعی نیز دارد که آن را به یک زبان همه‌منظوره برای توسعه انواع برنامه‌ها تبدیل می‌کند.

تازه‌های C# 7

در C# 6 که منبع کامپایلر به‌صورت کامل باز نهایی شد. علاوه بر آن خط لوله کامپایلر جدید می‌تواند از طریق برنامه‌های سفارشی مورد استفاده قرار گیرد؛ بنابراین بهبود کامپایلر بسیار ساده‌تر می‌شود.

C# 7 نیز ویژگی‌های جدیدی را معرفی کرده ریش در پس‌زمینه برنامه‌نویسی تابعی داشتند؛ مانند توابع محلی، تاپل‌ها و مقایسه الگوها که در بخش بعدی کتاب به آنها خواهیم پرداخت. تازه‌های C# 7 به اختصار عبارت‌اند از:

- جداکننده‌های ارقام؛
- لیترال‌های دودویی؛
- اعضای Expression-Bodied؛
- متغیر out؛
- آرگومان‌های نامدار بدون دنباله؛
- ساختارهای فقط خواندنی؛
- پارامترهای in؛
- دارنده دسترسی private protected؛
- نوع هدف پیش‌فرض؛
- توابع محلی؛
- تاپل‌ها؛
- نام‌های تاپل استنتاج‌شده؛
- Destructorها؛
- مقایسه الگوها؛

■ ارسال عبارات؛

■ Async Main

■ معناشناسی ارجاع‌ها.

در فصل‌های مختلف کتاب به بیشتر این مطالب جدید خواهیم پرداخت.

تازه‌های ASP.NET CORE

Visual Studio 2017 و Visual Studio 2015 یک فایل پروژه جدید دارند. ابزارهای NET Core. که در پیش‌نمایش 2015 Visual Studio بودند، به همراه Visual Studio 2017 منتشر شدند. این ابزارها با فایل‌های csproj به محیط MSBuild منتقل می‌شدند؛ بنابراین اکنون فایل‌های csproj را در برنامه‌های NET Framework و NET Core داریم. البته این فایل‌ها همان فایل‌های csproj نیستند که از نسل‌های پیش با آنها آشنا بودید. فایل‌های csproj بسیار کوتاه‌تر و ساده‌تر شده‌اند و نیز می‌توانید آنها را با استفاده از یک ویرایشگر متن ساده تغییر دهید.

NET Core 2.0 با تغییراتی در تعریف‌شده در NET Standard 2.0. بهبود یافته است که این کار باعث می‌شود منتقل کردن برنامه‌های NET Framework به NET Core آسان‌تر شود.

با ایجاد یک پروژه NET Core، تنها فایل‌های csproj، بلکه کدهای C# نیز ساده‌تر شده‌اند. هنگامی که از WebHostBuilder استفاده می‌کنید، خیلی موارد از پیش تعریف می‌شوند. فراهم‌کننده‌های پیکربندی و ثبت به‌طور خودکار و بدون نیاز به افزودن از طرف شما افزوده می‌گردند. با ASP.NET Core MVC نیز تغییرات کوچکی ایجاد شده است؛ برای مثال شما می‌توانید با استفاده از کمک‌کننده tag مورد استفاده قرار گیرید.

همچنین یک فناوری جدید (Razor Pages) وجود دارد که یادگیری MVC در ASP.NET Core را ساده‌تر می‌کند. برخی برنامه‌ها نیاز به انتزاع از الگوی Model-View-Controller ندارند. اینجا جایی است که Razor Pages جایگاه خود را می‌یابد.

تازه‌های پلتفرم عمومی (UWP) Windows

هر سال دوبار به‌نگام‌سازی‌های Windows 10 عرضه می‌شود (برای کاربران Windows باشید، به‌نگام‌سازی‌های بیشتری دریافت می‌کنید؛ ولی برای تمام کاربران چنین نیست). با هر به‌نگام‌سازی Windows یک SDK جدید منتشر می‌شود.

مایکروسافت به‌ارایه طراحی ویژگی‌های جدید ادغام‌شده در کنترل‌های Windows ادامه می‌دهد. طراحی جدید Fluent Design نامیده می‌شود که در کنترل‌های استاندارد ادغام شده است و به‌طور مستقیم نیز در Parallax View برای افزودن افکت parallax به برنامه‌ها افزوده شده است.

ویژگی‌هایی نیز برای بهبود تولید افزوده شده‌اند. می‌توانید از Windows Template Studio استفاده کنید (که به‌عنوان یک گسترش Visual Studio افزوده شده است) تا یک ویرایشگر الگو داشته باشید و صفحات بیشتری تولید کنید و از سرویس‌های از پیش تولیدشده استفاده نمایید.

XAML با XAML شرطی بهبود داده شده است که از نسخه‌های مختلف Windows 10 پشتیبانی کند و ویژگی‌هایی را به‌کاربرد که در نسخه‌های قدیمی Windows 10 در دسترس نیستند.

کنترل InkCanvas خط‌کش‌های جدیدی عرضه می‌کند که به‌سادگی می‌توانند در برنامه‌های شما ادغام شوند. NavigationView ساخت منوهای تطبیق‌پذیر را با یک دکمه hamburger و یک SplitView ساده‌تر می‌کند [این مطالب در بخش چهارم کتاب زبان اصلی ارائه شده‌اند، ولی در ترجمه فارسی به آنها نپرداخته‌ایم].

برای نوشتن و اجرای کدهای C# به چه مواردی نیاز دارید

NET Core. روی سیستم‌عامل‌های Windows، Linux و Mac اجرا می‌شود. با استفاده از Visual Studio Code می‌توان روی هر کدام از این سیستم‌عامل‌ها برنامه‌ها را ایجاد و اجرا کرد (<https://code.visualstudio.com>). بهترین توسعه‌دهنده برای استفاده و توسعه‌دهنده‌ای که در این کتاب مورد استفاده قرار گرفته، Visual Studio 2017 است. می‌توانید از نسخه Visual Studio Community 2017 به نشانی <https://www.visualstudio.com> استفاده کنید؛ ولی برخی ویژگی‌های مطرح‌شده در کتاب، تنها در نسخه Visual Studio Enterprise در دسترس‌اند. البته هر جا که به نسخه Enterprise نیاز باشد، در متن کتاب ذکر خواهد شد. Visual Studio 2017 برای نصب نیاز به Windows 10 build 1507 یا بالاتر، Windows 8.1، Windows Server 2012 R2 یا Windows 7 SP1 دارد. برای ساخت و اجرای برنامه‌های (UWP) Windows مطرح‌شده در کتاب به Windows 10 نیاز دارید.

برای ایجاد و ساخت برنامه‌های Xamarin برای محیط iOS نیز نیاز به Mac برای سیستم ساخت دارید. بدون Mac شما می‌توان برنامه‌های Xamarin را برای Windows و Android ایجاد کرد.

برای توسعه برنامه با Mac نیز می‌توانید از Visual Studio مخصوص Mac به نشانی <https://www.visualstudio.com/visual-studio-mac/> استفاده کنید. همچنین می‌توان از این ابزار برای ایجاد برنامه‌های ASP.NET Core و Xamarin استفاده کرد؛ ولی نمی‌توان با آن برنامه Windows را ایجاد و آزمایش نمود.

این کتاب چه مطالبی را پوشش می‌دهد

این کتاب با مروری بر معماری کلی .NET، اصل، اول آغاز می‌شود تا پیش‌زمینه لازم برای نوشتن کدهای مدیریت‌شده را آرایه کند. در این فصل خلاصه‌ای درباره انواع مختلف برنامه‌ها می‌خوانید و با چگونگی کامپایل در محیط توسعه جدید CLI آشنا می‌شوید. بخش‌های مهم و لازم برای شروع کار در Visual Studio را مشاهده می‌کنید. پس از آن، کتاب به بخش‌هایی تقسیم شده است که زبان C# و کاربرد آن در زمینه‌های متنوع را پوشش دهد.

در ادامه به اصول زبان C# می‌پردازیم. این بخش نیاز به دانش قبلی درباره هیچ زبان دیگری ندارد و فرض می‌کنیم که شما برنامه‌نویس باتجربه‌ای هستید. کار با مروری بر ساختارهای اصلی زبان C# و انواع داده‌ها آغاز می‌شود. سپس به شیءگرایی پرداخته و در ادامه به مباحث پیشرفته‌تر C# مانند generics، عملگرها، آرایه‌ها، نماینده‌ها، عبارات لمبدا، رشته‌ها و عبارات منظم و کوثری ادغام‌شده در زبان (LINQ) پرداخته‌ایم.

چون خیلی از ویژگی‌های C# ریشه در برنامه‌نویسی تابعی دارند، در یک فصل مستقل با برنامه‌نویسی تابعی و تایلرها آشنا می‌شوید. سپس به خطاها و مدیریت استثناها پرداخته‌ایم و در ادامه در یک فصل مجزا، محیط Visual Studio 2017 را بررسی نموده‌ایم. فایل‌ها و استریم‌ها، مباحث مربوط به ADO.NET و تراکنش‌ها و Entity Framework Core عنوان‌های فصل‌های پایانی کتاب هستند.

توجه

این کتاب با هدف آموزش کامل و آسان، لبریز از مثال‌ها و کدهای مختلف است. مطالب هر فصل با استفاده از مثال‌های گوناگون آموزش داده شده است. برای داشتن همه کدهای این مثال‌ها کافی است به وبسایت ناشر (آدرس www.kianpub.com) مراجعه کنید و پس از جست‌وجوی نام کتاب، از قسمت «فایل‌های مرتبط» در صفحه کتاب کدهای مثال را دانلود کنید.

فصل اول: برنامه‌ها و ابزارهای .NET

۳۱	۱-۱. انتخاب فناوری مناسب.....
۳۱	۲-۱. مروری بر تاریخچه .NET.....
۳۲	۱-۲-۱. C# 1.0 - یک زبان جدید.....
۳۴	۲-۲-۱. C# 2 و .NET 2 با Generic ها.....
۳۴	۳-۲-۱. .NET 3، پی‌ریزی رایانه ویندوز (WPF).....
۳۵	۴-۲-۱. C# 3 و .NET 3.5، LINQ.....
۳۶	۵-۲-۱. C# 4 و .NET 4، پویایی و TPL.....
۳۶	۶-۲-۱. C# 5 برنامه‌نویسی ناهمگام.....
۳۷	۷-۲-۱. .NET Core 1.0، C# 5.....
۳۸	۸-۲-۱. .NET Core 2.0 و C# 7.....
۳۹	۳-۱. انتخاب فناوری حرکت به جلو.....
۴۰	۴-۱. اصطلاحات .NET.....
۴۱	۱-۴-۱. .NET Framework.....
۴۱	۲-۴-۱. .NET Core.....
۴۲	۳-۴-۱. .NET Standard.....
۴۲	۵-۱. بسته‌های NuGet.....
۴۳	۶-۱. فضاهای نام.....
۴۴	۷-۱. زبان زمان اجرای مشترک.....
۴۵	۸-۱. Windows Runtime.....
۴۶	۹-۱. استفاده از واسط خط فرمان .NET CORE (CLI).....
۴۶	۱-۹-۱. نصب محیط.....
۴۷	۲-۹-۱. ایجاد برنامه.....
۴۹	۳-۹-۱. ساخت برنامه.....
۴۹	۴-۹-۱. اجرای برنامه.....
۵۰	۵-۹-۱. ایجاد یک برنامه وب.....
۵۱	۶-۹-۱. انتشار برنامه.....
۵۲	۷-۹-۱. انتشارهای خودمختار.....
۵۲	۱۰-۱. استفاده از Visual Studio 2017.....
۵۳	۱-۱۰-۱. نصب Visual Studio 2017.....
۵۳	۲-۱۰-۱. ایجاد پروژه.....
۵۴	۳-۱۰-۱. کار با Solution Explorer.....
۵۵	۴-۱۰-۱. پیکربندی مشخصات پروژه.....

۵۶	۵-۱۰-۱. آشنایی با ویرایشگر.....
۵۷	۶-۱۰-۱. ساخت پروژه.....
۵۷	۷-۱۰-۱. اجرای برنامه.....
۵۸	۸-۱۰-۱. اشکال‌زدایی.....
۵۸	۱۱-۱۰-۱. انواع برنامه‌ها و فناوری‌ها.....
۵۹	۱-۱۱-۱. دسترسی به داده.....
۵۹	۲-۱۱-۱. برنامه‌های Windows.....
۵۹	۳-۱۱-۱. Xamarin.....
۶۰	۴-۱۱-۱. برنامه‌های وب.....
۶۰	۵-۱۱-۱. Web API.....
۶۰	۶-۱۱-۱. SignalR و WebHook.....
۶۰	۷-۱۱-۱. Microsoft Azure.....
۶۲	۱۲-۱. ابزارهای توسعه دهندگان.....
۶۲	۱-۱۲-۱. Visual Studio Community.....
۶۲	۲-۱۲-۱. Visual Studio Professional.....
۶۲	۳-۱۲-۱. Visual Studio Enterprise.....
۶۳	۴-۱۲-۱. Visual Studio برای Mac.....
۶۳	۵-۱۲-۱. Visual Studio Code.....
۶۳	۱۳-۱. خلاصه.....

فصل دوم: Core C#

۶۷	۱-۲. مبانی C#.....
۶۷	۱-۱-۲. Hello, World!.....
۶۹	۲-۲. کار با متغیرها.....
۷۰	۱-۲-۲. مقدارگذاری اولیه متغیرها.....
۷۰	۲-۲-۲. استفاده از استنتاج نوع.....
۷۱	۳-۲-۲. آشنایی با محدوده متغیرها.....
۷۴	۴-۲-۲. کار با ثابت‌ها.....
۷۵	۳-۲. استفاده از انواع داده‌ای از پیش تعریف‌شده.....
۷۵	۱-۳-۲. انواع مقادیر و ارجاع‌ها.....
۷۶	۲-۳-۲. نوع‌های NET.....
۷۷	۳-۳-۲. نوع‌های مقدار از پیش تعریف‌شده.....
۸۳	۴-۲. کنترل جریان برنامه.....
۸۳	۱-۴-۲. عبارات شرطی.....

۸۷	 ۲-۴-۲. حلقه‌ها
۹۱	 ۳-۴-۲. عبارات پرش
۹۱	 ۵-۲. سازمان‌دهی با فضا‌های نام
۹۳	 ۱-۵-۲. کاربرد دستورالعمل
۹۴	 ۲-۵-۲. نام‌های مستعار فضای نام
۹۵	 ۶-۲. آشنایی با متد Main
۹۶	 ۷-۲. استفاده از توضیحات
۹۶	 ۱-۷-۲. توضیحات داخلی درون فایل‌های منبع کد
۹۷	 ۲-۷-۲. مستندسازی XMI
۹۸	 ۸-۲. آشنایی با پیش‌پردازنده دستورالعمل‌های C#
۹۸	 ۹-۲. خطوط راهنمای برنامه‌نویسی C#
۹۹	 ۱-۹-۲. قواعد شناسه‌ها
۱۰۰	 ۱۰-۲. خلاصه

فصل سوم: اشیاء و نوع‌ها

۱۰۳	 ۱-۳. ایجاد و استفاده از کلاس‌ها
۱۰۳	 ۲-۳. کلاس‌ها و ساختارها
۱۰۴	 ۳-۳. کلاس‌ها
۱۰۵	 ۱-۳-۳. فیلدها
۱۱۲	 ۲-۳-۳. متدها
۱۱۷	 ۳-۳-۳. سازنده‌ها
۱۲۳	 ۴-۳. ساختارها
۱۲۵	 ۱-۴-۳. ساختارها نوع داده‌اند
۱۲۶	 ۲-۴-۳. ساختارهای فقط خواندنی
۱۲۶	 ۳-۴-۳. ساختارها و وراثت
۱۲۷	 ۴-۴-۳. سازنده‌هایی برای ساختارها
۱۲۷	 ۵-۴-۳. ساختارهای ارجاعی
۱۲۸	 ۵-۲. ارسال پارامترها با مقدار و ارجاع
۱۲۹	 ۱-۵-۳. پارامترهای ارجاعی
۱۳۰	 ۲-۵-۳. پارامترهای out
۱۳۱	 ۳-۵-۳. پارامترهای in
۱۳۲	 ۶-۳. نوع‌های تهی‌شدنی
۱۳۳	 ۷-۳. نوع‌های شمارشی
۱۳۵	 ۸-۳. کلاس‌های جزئی

۱۳۸	۹-۳. متدهای گسترش.....
۱۳۹	۱۰-۳. کلاس Object.....
۱۴۰	۱۱-۳. خلاصه.....

فصل چهارم: برنامه‌نویسی شیء‌گرا با C#

۱۴۲	۱-۴. شیء‌گرایی.....
۱۴۳	۲-۴. انواع وراثت.....
۱۴۳	۱-۲-۴. وراثت چندگانه.....
۱۴۴	۲-۲-۴. ساختارها و کلاس‌ها.....
۱۴۴	۳-۴. بسته‌سازی و وراثت.....
۱۴۵	۱-۱-۴. متدهای مجازی.....
۱۴۷	۲-۳-۴. چرخش.....
۱۴۸	۳-۳-۴. پنهان کردن.....
۱۴۹	۴-۳-۴. فراخوانی مجدد.....
۱۵۰	۵-۳-۴. متدها و کلاس‌های انزاعی.....
۱۵۱	۶-۳-۴. کلاس‌ها و متدهای عمومی.....
۱۵۲	۷-۳-۴. سازنده‌های کلاس‌های مشتق شده.....
۱۵۵	۴-۴. تغییردهنده‌ها.....
۱۵۵	۱-۴-۴. تغییردهنده‌های دسترسی.....
۱۵۶	۲-۴-۴. تغییردهنده‌های دیگر.....
۱۵۷	۵-۴. واسطه‌ها.....
۱۵۸	۱-۵-۴. تعریف و پیاده‌سازی واسطه‌ها.....
۱۶۱	۲-۵-۴. وراثت واسطه‌ها.....
۱۶۳	۶-۴. عملگرهای is و as.....
۱۶۴	۷-۴. خلاصه.....

فصل پنجم: Generic ها

۱۶۷	۱-۵. مروری بر generic ها.....
۱۶۷	۱-۱-۵. کارایی.....
۱۶۹	۲-۱-۵. ایمنی نوع.....
۱۶۹	۳-۱-۵. استفاده مجدد از کد دودویی.....
۱۷۰	۴-۱-۵. حجیم شدن کد.....
۱۷۰	۵-۱-۵. خطوط راهنمای نام‌گذاری.....
۱۷۰	۲-۵. ایجاد کلاس‌های generic.....

۱۷۴	۳-۵. ویژگی‌های generic
۱۷۵	۱-۳-۵. مقادیر پیش‌فرض
۱۷۵	۲-۳-۵. محدودیت‌ها
۱۷۸	۳-۳-۵. وراثت
۱۷۹	۴-۳-۵. اعضای ایستا
۱۷۹	۴-۵. واسطه‌های generic
۱۸۰	۱-۴-۵. کواریانس و کونترا واریانس
۱۸۱	۲-۴-۵. کواریانس با واسطه‌های generic
۱۸۲	۲-۴-۵. Contra-Variance با واسطه‌های generic
۱۸۳	۵-۵. سایر باره‌های generic
۱۸۶	۶-۵. متدهای generic
۱۸۶	۱-۶-۵. مثالی از متدهای generic
۱۸۷	۲-۶-۵. متدهای generic با مژده
۱۸۸	۳-۶-۵. متدهای generic با Delegate
۱۸۹	۴-۶-۵. تخصیص‌سازی متدهای generic
۱۹۰	۷-۵. خلاصه

فصل ششم: عملگرها و castها

۱۹۳	۱-۶. عملگرها و castها
۱۹۳	۲-۶. عملگرها
۱۹۵	۱-۲-۶. میانبرهای عملگرها
۱۹۶	۲-۲-۶. عملگر عبارت شرطی (?:)
۱۹۶	۳-۲-۶. عملگرهای checked و unchecked
۱۹۷	۴-۲-۶. عملگر is
۱۹۸	۵-۲-۶. عملگر as
۱۹۹	۶-۲-۶. عملگر sizeof
۱۹۹	۷-۲-۶. عملگر typeof
۱۹۹	۸-۲-۶. عملگر nameof
۲۰۰	۹-۲-۶. عملگر شاخص
۲۰۱	۱۰-۲-۶. نوع‌های تهی‌شدنی و عملگرها
۲۰۲	۱۱-۲-۶. عملگر تلفیق تهی
۲۰۳	۱۲-۲-۶. عملگر شرطی تهی
۲۰۴	۱۳-۲-۶. اولویت عملگرها و شرکت‌پذیری
۲۰۵	۳-۶. ایمنی نوع

۲۰۵	۱-۳-۶. تبدیل نوع.....
۲۱۰	۲-۳-۶. Boxing and Unboxing.....
۲۱۰	۴-۶. مقایسه برابری اشیا.....
۲۱۱	۱-۴-۶. بررسی برابری نوع‌های ارجاع.....
۲۱۲	۲-۴-۶. مقایسه برابری نوع‌های مقدار.....
۲۱۳	۵-۶. بیش‌بارگذاری عملگرها.....
۲۱۴	۱-۵-۶. چگونگی کارکرد عملگرها.....
۲۱۵	۲-۵-۶. مثالی از بیش‌بارگذاری عملگر: ساختار Vector.....
۲۱۹	۳-۵-۶. بیش‌بارگذاری عملگرهای مقایسه.....
۲۲۱	۴-۵-۶. خدایه عملگرها را می‌توان بیش‌بارگذاری کرد؟.....
۲۲۲	۶-۶. پیاده‌سازی عملگر سفارشی Index.....
۲۲۴	۷-۶. cast. تعریف شده وسیله کاربر.....
۲۲۶	۸-۶. خلاصه.....

فصل هفتم: آرایه‌ها

۲۲۹	۱-۷. چند شیء از نوع یکسان.....
۲۲۹	۲-۷. آرایه‌های ساده.....
۲۲۹	۱-۲-۷. تعریف آرایه.....
۲۲۹	۲-۲-۷. مقدارگذاری آرایه.....
۲۳۰	۳-۲-۷. دسترسی به عناصر آرایه.....
۲۳۱	۴-۲-۷. استفاده از نوع‌های ارجاع.....
۲۳۲	۳-۷. آرایه‌های چندبعدی.....
۲۳۳	۴-۷. آرایه‌های نامموار.....
۲۳۴	۵-۷. کلاس Array.....
۲۳۵	۱-۵-۷. ایجاد آرایه‌ها.....
۲۳۶	۲-۵-۷. کپی کردن آرایه‌ها.....
۲۳۷	۳-۵-۷. مرتب‌سازی.....
۲۴۰	۶-۷. آرایه‌ها به عنوان پارامتر.....
۲۴۰	۷-۷. کواریانس آرایه‌ها.....
۲۴۰	۸-۷. شمارنده‌ها.....
۲۴۱	۱-۸-۷. واسط Enumerator.....
۲۴۱	۲-۸-۷. عبارت foreach.....
۲۴۲	۳-۸-۷. عبارت yield.....
۲۴۴	۴-۸-۷. روش‌های مختلف برای حرکت درون مجموعه‌ها.....

۲۴۵	۵-۸-۷. بازگرداندن شمارنده‌ها با yield return
۲۴۷	۹-۷. مقایسه ساختاری
۲۴۸	۱۰-۷. Spans
۲۴۹	۱۰-۷. ایجاد برش‌ها
۲۵۰	۱۰-۷. تغییر مقادیر با استفاده از span
۲۵۱	۱۰-۷. span‌های فقط خواندنی
۲۵۲	۱۱-۷. مخزن آرایه
۲۵۲	۱۱-۷. ایجاد مخزن آرایه
۲۵۲	۱۱-۷. آچار حافظه از مخزن
۲۵۲	۱۱-۷. بازگرداندن حافظه به مخزن
۲۵۳	۱۲-۷. خلاصه

فصل هشتم: نماینده‌ها، عبارات لمبدا و رویدادها

۲۵۷	۱-۸. متدهای ارجاع‌دهنده
۲۵۷	۲-۸. نماینده‌ها
۲۵۸	۱-۲-۸. تعریف نماینده‌ها
۲۵۹	۲-۲-۸. استفاده از نماینده‌ها
۲۶۲	۳-۲-۸. مثال ساده از نماینده‌ها
۲۶۴	۴-۲-۸. نماینده‌های Action<T> و Func<T>
۲۶۵	۵-۲-۸. مثال BubbleSorter
۲۶۸	۶-۲-۸. نماینده‌های چندپخشی
۲۷۲	۷-۲-۸. متدهای ناشناس
۲۷۳	۳-۸. عبارات لمبدا
۲۷۳	۱-۳-۸. پارامترها
۲۷۴	۲-۳-۸. خطوط کد چندگانه
۲۷۴	۳-۳-۸. بستارها
۲۷۵	۴-۸. رویدادها
۲۷۶	۱-۴-۸. منتشرکننده رویداد
۲۷۸	۲-۴-۸. شنونده‌ی رویداد
۲۷۹	۵-۸. خلاصه

فصل نهم: رشته‌ها و عبارات منظم

۲۸۳	۱-۹. بررسی System.String
۲۸۵	۱-۱-۹. ساخت رشته‌ها

۲۸۸	۲-۱-۹. اعضای StringBuilder
۲۸۹	۲-۹. قالب‌های رشته
۲۸۹	۱-۲-۹. الحاق رشته‌ها
۲۹۰	۲-۲-۹. FormattableString
۲۹۰	۳-۲-۹. استفاده از زبان‌های دیگر با الحاق رشته‌ها
۲۹۱	۴-۲-۹. پرش از آکولاد باز و بسته ({})
۲۹۱	۵-۲-۹. قالب‌های DateTime و اعداد
۲۹۳	۶-۲-۹. قالب‌های رشته سفارشی
۲۹۴	۳-۹. عبارات منظم
۲۹۴	۱-۳-۹. ارفق عبارات منظم
۲۹۶	۲-۳-۹. مثال RegularExpressionsPlayground
۳۰۰	۳-۳-۹. نمایش نتایج
۳۰۱	۴-۳-۹. همخوانی با، گام‌ها و پوشش‌ها
۳۰۴	۴-۹. رشته‌ها و محدودیت‌ها
۳۰۵	۵-۹. خلاصه

فصل دهم: مجموعه‌ها

۳۰۹	۱-۱۰. مرور
۳۰۹	۲-۱۰. واسط‌های مجموعه و نوع‌ها
۳۱۰	۳-۱۰. فهرست‌ها
۳۱۲	۱-۳-۱۰. ایجاد فهرست‌ها
۳۱۲	۲-۳-۱۰. مقدارگذاری‌کننده مجموعه
۳۱۳	۳-۳-۱۰. افزودن عناصر
۳۱۴	۴-۳-۱۰. درج عناصر
۳۱۴	۵-۳-۱۰. دستیابی به عناصر
۳۱۵	۶-۳-۱۰. حذف عناصر
۳۱۵	۷-۳-۱۰. جست‌وجو
۳۱۷	۸-۳-۱۰. مرتب‌سازی
۳۲۰	۹-۳-۱۰. مجموعه‌های فقط خواندنی
۳۲۰	۴-۱۰. صف‌ها
۳۲۴	۵-۱۰. پشته‌ها
۳۲۶	۶-۱۰. لیست‌های پیوندی
۳۳۱	۷-۱۰. فهرست‌های (لیست‌های) مرتب‌شده
۳۳۳	۸-۱۰. فرهنگ‌های لغت

۳۳۴	۱-۸-۱۰. مقدارگذاری کننده اولیه فرهنگ لغت
۳۳۴	۲-۸-۱۰. نوع کلید
۳۳۶	۳-۸-۱۰. مثال فرهنگ لغت
۳۴۰	۴-۸-۱۰. جست و جوها
۳۴۰	۵-۸-۱۰. فرهنگ های لغت ذخیره شده
۳۴۱	۹-۱۰. Laset
۳۴۲	۱۰-۱۰. کارایی
۳۴۴	۱۱-۱۰. خلاصه

فصل یازدهم: کوثری ادغام شده در زبان

۳۴۷	۱-۱۱. مرور بر LINQ
۳۴۷	۱-۱-۱۱. فهرست ها و وجودیت ها
۳۵۱	۲-۱-۱۱. کوثری LINQ
۳۵۲	۳-۱-۱۱. متدهای گسترش
۳۵۴	۴-۱-۱۱. اجرای کوثری تعلیق
۳۵۶	۲-۱۱. عملگرهای کوثری استاندارد
۳۵۸	۱-۲-۱۱. فیلتر کردن
۳۵۹	۲-۲-۱۱. فیلتر کردن با شاخص
۳۶۰	۳-۲-۱۱. فیلتر کردن نوع
۳۶۰	۴-۲-۱۱. from مرکب
۳۶۲	۵-۲-۱۱. مرتب سازی
۳۶۳	۶-۲-۱۱. دسته بندی
۳۶۵	۷-۲-۱۱. متغیرهای درون کوثری LINQ
۳۶۶	۸-۲-۱۱. دسته بندی اشیای تودرتو
۳۶۷	۹-۲-۱۱. پیوند داخلی
۳۷۰	۱۰-۲-۱۱. پیوند خارجی از چپ
۳۷۲	۱۱-۲-۱۱. پیوند گروهی
۳۷۶	۱۲-۲-۱۱. عملگرهای set
۳۷۷	۱۳-۲-۱۱. Zip
۳۷۸	۱۴-۲-۱۱. افراز
۳۸۰	۱۵-۲-۱۱. عملگرهای توده ساز
۳۸۱	۱۶-۲-۱۱. عملگرهای تبدیل
۳۸۳	۱۷-۲-۱۱. عملگرهای تولید
۳۸۴	۲۰-۱۱. LINQ موازی

۳۸۴	۱-۳-۱۱. کوئری‌های موازی
۳۸۵	۲-۳-۱۱. افزازکننده‌ها
۳۸۶	۳-۳-۱۱. لغو
۳۸۷	۴-۳-۱۱. درخت‌های عبارت
۳۹۰	۵-۳-۱۱. فراهم‌کننده‌های LINQ
۳۹۰	۶-۳-۱۱. خلاصه

فصل دوازدهم: برنامه‌نویسی تابعی با C#

۳۹۵	۱-۱۲. برنامه‌نویسی تابعی چیست؟
۳۹۵	۱-۱-۱۲. جتناب از تغییر وضعیت
۳۹۶	۲-۱-۱۲. تبع به‌عنوان کلاس‌های نخست
۳۹۶	۳-۱-۱۲. توابع مرتبه بالاتر
۳۹۷	۴-۱-۱۲. توابع نالص
۳۹۷	۲-۱۲. اعضای EXPRESSION BODIED
۳۹۸	۳-۱۲. متدهای گسترش
۳۹۹	۴-۱۲. استفاده از static
۴۰۲	۵-۱۲. توابع محلی
۴۰۳	۱-۵-۱۲. توابع محلی با عبارت yield
۴۰۶	۲-۵-۱۲. توابع محلی بازگشتی
۴۰۷	۶-۱۲. تاپل‌ها
۴۰۸	۱-۶-۱۲. تعریف و مقدارگذاری اولیه تاپل‌ها
۴۰۹	۲-۶-۱۲. شکستن ساختار تاپل
۴۱۰	۳-۶-۱۲. بازگرداندن تاپل‌ها
۴۱۰	۴-۶-۱۲. پشت‌صحنه
۴۱۲	۵-۶-۱۲. سازگاری Value Tuple با Tuple
۴۱۳	۶-۶-۱۲. استنتاج نام تاپل‌ها
۴۱۴	۷-۶-۱۲. تاپل‌هایی با لیست‌های پیوندی
۴۱۴	۸-۶-۱۲. تاپل‌ها با LINQ
۴۱۵	۹-۶-۱۲. شکستن ساختار
۴۱۶	۱۰-۶-۱۲. شکستن ساختار با متدهای گسترش
۴۱۷	۷-۱۲. مقایسه الگوها
۴۱۸	۱-۷-۱۲. مقایسه الگو با عملگر is
۴۲۰	۲-۷-۱۲. مقایسه الگو با استفاده از عبارت switch
۴۲۱	۳-۷-۱۲. مقایسه الگوها با generic

۴۲۲	۸-۱۲. خلاصه
-----	-------------

فصل سیزدهم: خطاها و استثناها

۴۲۵	۱-۱۳. مقدمه
۴۲۵	۲-۱۳. کلاس‌های استثنا
۴۲۷	۳-۱۳. گرفتن استثناها
۴۳۰	۱-۳-۱۳. استثناها و کارایی
۴۳۱	۲-۳-۱۳. پیاده‌سازی بلوک‌های catch چندگانه
۴۳۵	۳-۳-۱۳. گرفتن استثناها از یک کد دیگر
۴۳۵	۴-۳-۱۳. System.Exception
۴۳۶	۵-۳-۱۳. فیلترهای استثناها
۴۳۷	۶-۳-۱۳. گسیل مجدد استثناها
۴۳۹	۷-۳-۱۳. استفاده برمی‌گردد از بایستی استثنا
۴۳۹	۸-۳-۱۳. تغییر استثنا
۴۴۰	۹-۳-۱۳. گسیل مجدد استثنا
۴۴۱	۱۰-۳-۱۳. استفاده از فیلترها برای ورودی‌های
۴۴۲	۱۱-۳-۱۳. اگر استثنایی مدیریت نشود، چه اتفاقی می‌افتد؟
۴۴۲	۴-۱۳. کلاس‌های استثنای تعریف‌شده به‌وسیله
۴۴۳	۱-۴-۱۳. گرفتن استثناهای تعریف‌شده به‌وسیله
۴۴۵	۲-۴-۱۳. گسیل استثناهای تعریف‌شده به‌وسیله
۴۴۹	۳-۴-۱۳. تعریف کلاس‌های استثنای تعریف‌شده به‌وسیله
۴۵۱	۵-۱۳. اطلاعات فراخوانی‌کننده
۴۵۲	۶-۱۳. خلاصه

فصل چهاردهم: Visual Studio 2017

۴۵۵	۱-۱۴. کار با Visual Studio 2017
۴۵۹	۱-۱-۱۴. نسخه‌های Visual Studio
۴۵۹	۲-۱-۱۴. تنظیمات Visual Studio
۴۶۰	۲-۱۴. ایجاد یک پروژه
۴۶۱	۱-۲-۱۴. NET با چند هدف
۴۶۳	۲-۲-۱۴. انتخاب نوع پروژه
۴۶۴	۳-۲-۱۴. استفاده از الگوهای Windows Universal Project
۴۶۵	۴-۲-۱۴. استفاده از الگوهای پروژه NET Core
۴۶۷	۵-۲-۱۴. استفاده از الگوهای استاندارد NET

۴۶۷	۳-۱۴. کشف و کدنویسی یک پروژه
۴۶۷	۱-۳-۱۴. Solution Explorer
۴۷۶	۲-۳-۱۴. کار با ویرایشگر کد
۴۸۲	۳-۳-۱۴. آموزش و درک پنجره‌های دیگر
۴۸۸	۴-۳-۱۴. چیدمان پنجره‌ها
۴۸۹	۴-۱۴. ساخت یک پروژه
۴۸۹	۱-۴-۱۴. ساخت، کامپایل و make
۴۸۹	۲-۴-۱۴. اشکال‌زدایی و آزادسازی buildها
۴۹۰	۳-۴-۱۴. نه‌سازی
۴۹۰	۴-۴-۱۴. نمایش اشکال‌زدا
۴۹۱	۵-۴-۱۴. استورات اشکال‌زدایی کد اضافی
۴۹۱	۶-۴-۱۴. انتخاب یک پیکر
۴۹۲	۷-۴-۱۴. ویرایش یک بدو
۴۹۴	۵-۱۴. اشکال‌زدایی کد
۴۹۴	۱-۵-۱۴. تنظیم نقاط گس
۴۹۵	۲-۵-۱۴. استفاده از Data Tip ها و ویژالایز اشکال‌زدا
۴۹۶	۳-۵-۱۴. درخت و ژوال زنده
۴۹۷	۴-۵-۱۴. پایش و تغییر متغیرها
۴۹۸	۵-۵-۱۴. استثناها
۴۹۹	۶-۵-۱۴. چندنخی
۵۰۰	۶-۱۴. ابزارهای بازسازی
۵۰۲	۷-۱۴. ابزارهای تشخیصی
۵۰۷	۸-۱۴. خلاصه

فصل پانزدهم: فایل‌ها و استریم‌ها

۵۱۱	۱-۱۵. مقدمه
۵۱۱	۲-۱۵. مدیریت سیستم فایل
۵۱۲	۱-۲-۱۵. بررسی اطلاعات درایو
۵۱۴	۲-۲-۱۵. کار با کلاس Path
۵۱۵	۳-۲-۱۵. ایجاد فایل‌ها و پوشه‌ها
۵۱۶	۴-۲-۱۵. دستیابی و تغییر خصوصیات فایل
۵۱۷	۵-۲-۱۵. استفاده از File برای خواندن و نوشتن
۵۱۹	۳-۱۵. شماره‌گذاری فایل‌ها
۵۲۰	۴-۱۵. کار با استریم‌ها

۵۲۲	۱-۴-۱۵. کار با استریم‌های فایل
۵۲۳	۲-۴-۱۵. ایجاد یک FileStream
۵۲۴	۳-۴-۱۵. دریافت اطلاعات استریم
۵۲۵	۴-۴-۱۵. تحلیل کدگذاری فایل متنی
۵۲۶	۵-۴-۱۵. خواندن استریم‌ها
۵۲۷	۶-۴-۱۵. نوشتن استریم‌ها
۵۲۸	۷-۴-۱۵. کپی کردن استریم‌ها
۵۲۹	۸-۴-۱۵. استفاده از استریم‌های با دسترسی تصادفی
۵۳۱	۹-۴-۱۵. استفاده از استریم‌های بافرشده
۵۳۱	۵-۱۵. استفاده از خواننده‌ها و نویسندگان
۵۳۲	۱-۵-۱۵. کلاس StreamReader
۵۳۳	۲-۵-۱۵. کلاس StreamWriter
۵۳۴	۳-۵-۱۵. خواندن و نوشتن فایل‌های باینری
۵۳۵	۶-۱۵. فشرده‌سازی فایل‌ها
۵۳۶	۱-۶-۱۵. استفاده از DeflateStream
۵۳۷	۲-۶-۱۵. استفاده از Brotli
۵۳۸	۳-۶-۱۵. zip کردن فایل‌ها
۵۳۸	۷-۱۵. مشاهده تغییرات فایل‌ها
۵۴۰	۸-۱۵. کار با فایل‌های نگاشت‌شده در حافظه
۵۴۱	۱-۸-۱۵. استفاده از دارندگان دسترسی برای ایجاد فایل‌های نگاشت‌شده در حافظه
۵۴۳	۲-۸-۱۵. استفاده از استریم‌ها برای ایجاد فایل‌های نگاشت‌شده در حافظه
۵۴۵	۹-۱۵. ارتباط با لوله‌ها
۵۴۶	۱-۹-۱۵. ایجاد یک سرویس‌دهنده لوله نامدار
۵۴۸	۲-۹-۱۵. ایجاد یک کلاینت لوله نامدار
۵۴۸	۳-۹-۱۵. ایجاد لوله‌های بی‌نام
۵۵۰	۱۰-۱۵. استفاده از فایل‌ها و استریم‌ها با Windows Runtime
۵۵۰	۱-۱۰-۱۵. برنامه ویرایشگر ویندوزی
۵۵۳	۲-۱۰-۱۵. نگاشت نوع‌های Windows Runtime به نوع‌های .NET
۵۵۵	۱۱-۱۵. خلاصه

فصل شانزدهم: ADO.NET و تراکنش‌ها

۵۵۹	۱-۱۶. مروری بر ADO.NET
۵۵۹	۱-۱۶. پایگاه داده نمونه
۵۵۹	۲-۱-۱۶. بسته‌های NuGet و فضاها

۵۶۰	۲-۱۶. استفاده از اتصال‌های پایگاه داده
۵۶۲	۱-۲-۱۶. مدیریت رشته‌های اتصال
۵۶۲	۲-۲-۱۶. انتلاف‌های اتصال
۵۶۳	۳-۲-۱۶. اطلاعات اتصال
۵۶۵	۳-۱۶. فرمان‌ها
۵۶۶	۱-۳-۱۶. ExecuteNonQuery
۵۶۷	۲-۳-۱۶. ExecuteScalar
۵۶۸	۳-۳-۱۶. ExecuteReader
۵۷۰	۴-۳-۱۶. فراخوانی رویه‌های ذخیره‌شده
۵۷۲	۴-۱۶. دسترسی ناهمگام به داده‌ها
۵۷۳	۵-۱۶. ترانزاکشن‌ها در ADO.NET
۵۷۸	۶-۱۶. تراکنش‌ها در System.Transactions
۵۸۱	۱-۶-۱۶. تراکنش‌ها، بل‌تثبیت
۵۸۲	۲-۶-۱۶. تراکنش‌ها، بسته‌ها
۵۸۵	۳-۶-۱۶. تراکنش‌های فراگیر
۵۸۷	۴-۶-۱۶. محدوده‌های تودرتو و تراکنش‌های فراگیر
۵۹۰	۷-۱۶. خلاصه

فصل هفدهم: Entity Framework Core

۵۹۳	۱-۱۷. معرفی EF Core
۵۹۴	۱-۱-۱۷. ایجاد مدل
۵۹۵	۲-۱-۱۷. قراردادهای یادآوری‌ها و fluent API
۵۹۵	۳-۱-۱۷. ایجاد زمینه
۵۹۶	۴-۱-۱۷. ایجاد پایگاه داده
۵۹۶	۵-۱-۱۷. حذف پایگاه داده
۵۹۷	۶-۱-۱۷. نوشتن در پایگاه داده
۵۹۸	۷-۱-۱۷. خواندن از پایگاه داده
۵۹۹	۸-۱-۱۷. بهنگام‌سازی رکوردها
۵۹۹	۹-۱-۱۷. حذف رکوردها
۶۰۰	۱۰-۱-۱۷. ثبت رویدادها
۶۰۱	۲-۱۷. استفاده از تزریق وابستگی
۶۰۴	۳-۱۷. ایجاد مدل
۶۰۴	۱-۳-۱۷. ایجاد یک ارتباط
۶۰۵	۲-۳-۱۷. یادآوری داده‌ها

۶۰۶	Fluent API. ۳-۳-۱۷
۶۰۸	پیکربندی نوع خودنگهدارنده. ۴-۳-۱۷
۶۰۹	داربست‌سازی یک مدل از پایگاه داده. ۵-۳-۱۷
۶۱۰	نگاشت به فیلدها. ۶-۳-۱۷
۶۱۲	خصوصیات سایه. ۷-۳-۱۷
۶۱۵	کوئری‌ها. ۴-۱۷
۶۱۵	کوئری‌های پایه. ۱-۴-۱۷
۶۱۷	ارزیابی کلاینت و سرویس‌دهنده. ۲-۴-۱۷
۶۱۹	کوئری‌های raw SQL. ۳-۴-۱۷
۶۲۰	روترهای کامپایل‌شده. ۴-۴-۱۷
۶۲۱	فیلترهای عمومی کوئری. ۵-۴-۱۷
۶۲۱	EF Functions. ۶-۴-۱۷
۶۲۲	ارتباط‌ها. ۵-۱۷
۶۲۲	ارتباط‌هایی که از قرارداد استفاده می‌کنند. ۱-۵-۱۷
۶۲۴	بارگذاری انحصاری داده‌ها- مرتب‌سازی. ۲-۵-۱۷
۶۲۶	بارگذاری مشتاقانه داده‌های مرتبط. ۳-۵-۱۷
۶۲۷	ارتباط‌هایی که از یادآوری‌ها استفاده می‌کنند. ۴-۵-۱۷
۶۲۸	ارتباط با استفاده از Fluent API. ۵-۵-۱۷
۶۳۰	جدول برای سلسله‌مراتب با قراردادها. ۶-۵-۱۷
۶۳۲	جدول برای سلسله‌مراتب با استفاده از Fluent API. ۷-۵-۱۷
۶۳۴	تقسیم جدول‌ها. ۸-۵-۱۷
۶۳۶	موجودیت‌های تصاحب‌شده. ۹-۵-۱۷
۶۳۸	ذخیره داده‌ها. ۶-۱۷
۶۳۸	افزودن اشیاء با ارتباط‌ها. ۱-۶-۱۷
۶۴۰	ردیابی شیء. ۲-۶-۱۷
۶۴۲	بهنگام‌سازی اشیاء. ۳-۶-۱۷
۶۴۲	بهنگام‌سازی اشیای ردیابی‌نشده. ۴-۶-۱۷
۶۴۳	دسته‌بندی. ۵-۶-۱۷
۶۴۵	مدیریت ناسازگاری. ۷-۱۷
۶۴۵	آخرین ذخیره‌کننده برنده است. ۱-۷-۱۷
۶۴۶	نخستین ذخیره‌کننده برنده است. ۲-۷-۱۷
۶۵۰	ائتلاف زمینه. ۸-۱۷
۶۵۱	استفاده از تراکنش‌ها. ۹-۱۷

۶۵۱.....	۱۷-۹-۱. استفاده از تراکنش‌های ضمنی.....
۶۵۴.....	۱۷-۹-۲. ایجاد صریح تراکنش‌ها.....
۶۵۵.....	۱۷-۱۰-۱. مهاجرت‌ها.....
۶۵۶.....	۱۷-۱۰-۱. آماده‌سازی فایل پروژه.....
۶۵۷.....	۱۷-۱۰-۲. میزبانی برنامه‌ها با ASP.NET Core MVC.....
۶۵۷.....	۱۷-۱۰-۳. میزبانی برنامه کنسولی .NET Core.....
۶۵۹.....	۱۷-۱۰-۴. ایجاد مهاجرت‌ها.....
۶۶۳.....	۱۷-۱۰-۵. اعمال مهاجرت‌ها از طریق برنامه‌نویسی.....
۶۶۴.....	۱۷-۱۰-۶. راه‌های دیگر برای اعمال مهاجرت‌ها.....
۶۶۴.....	۱۷-۱۱-۱. خلاصه.....