# کتاب آموزش سی شارپ به زبان فارسی
# (مقدماتی)

مناسب برای افراد مبتدی که می خواهند
بهترین زبان برنامه نویسی را فرا بگیرند

کدنویسی به زبان ساده
www.Youcode.ir

# فهرست

## سوم آشنایی با ویژوال استودیو



## چهارم آشنایی با انواع داده و متغیرها

## پنجم عبارت‌های محاسباتی



## ششم دستورهای شرطی



## هفتم دستورات تکرار (حلقه‌ها)

# فصل اول

# آشنایی با مفاهیم پایه‌ای پردازش داده‌ها

در این فصل ابتدا مفاهیم پایه‌ای پردازش داده‌ها شامل : داده‌ها، اطلاعات و پردازش که در کتاب مبانی کامپیوتر خوانده‌اید، یادآوری می‌شود و سپس در ادامه این فصل، با برنامه، زبان برنامه‌نویسی و تقسیم‌بندی زبان‌های برنامه‌نویسی از نظر نزدیکی به‌زبان محاوره‌ای و لزوم یادگیری یک زبان برنامه‌نویسی آشنا می‌شوید.

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

۱ــ مفهوم داده، اطلاعات و پردازش را بیان کند و آنها را به کار بندد.

۲ــ تعریف برنامه، برنامه‌نویسی، مترجم و انواع زبان‌های برنامه‌نویسی را از نظر سطح نزدیکی به زبان محاوره‌ای بیان کند.

۳ــ در یک برنامه ورودی، خروجی و پردازش را معین کند.

۴ــ هدف از برنامه‌نویسی را بیان نماید.

## ۱ــ۱ــ داده‌ها و اطلاعات[۱]

اغلب مردم دو واژه داده‌ها و اطلاعات را به یک معنی می‌دانند در صورتی که مفهوم و کاربرد این دو واژه با یکدیگر متفاوت است.

**داده‌ها**، مجموعه‌ای از مقادیر در مورد یک موضوع یا شیء است که به‌صورت کمّی یا یک مقدار عددی و یا به‌صورت کیفی نشان داده می‌شود.

مثلاً آزمون درس ریاضی یک کلاس، نمراتی حاصل می‌شوند که این نمرات به‌عنوان داده‌ها درنظر گرفته می‌شوند.

۱۷/۵ , ۱۴ , ۱۹ , ۱۱ , ۱۸ , ...

---

۱ــ Data & nformat on

اسامی دانش‌آموزان یک کلاس و یا نام شهرهای استان محل سکونت شما مثال‌های دیگری از داده ها می‌باشد.

در بعضی موارد نیز برای به‌دست آوردن داده‌های مربوط به یک موضوع، مجبور به استفاده از لوازم و وسایل مخصوص هستیم، مثلاً در مورد دمای محیط، نیاز به استفاده از یک حسگر[1] دما هستیم که به‌وسیلهٔ آن اندازه دما را به‌دست آوریم. اندازه دمای محیط، یک داده است.

برای اینکه از داده‌ها بتوانیم بهتر استفاده کنیم لازم است بر روی آنها محاسبات و یا به‌طور کلی **عملیاتی** را انجام دهیم. نتایج حاصل از این عملیات را **اطلاعات** می‌نامیم که می‌تواند مورد تفسیر و بررسی قرار گیرد و نتیجه بررسی آنها به **دانش** ختم گردد که دانش می‌تواند مبنای تصمیم‌گیری برای انجام کاری شود. مثلاً اگر داده‌ها را، نمرات ریاضی یک کلاس در نظر بگیریم و مجموع آنها را محاسبه و حاصل جمع را بر تعداد نمرات تقسیم کنیم، خارج قسمت به‌دست آمده میانگین یا معدل نمرات است. از روی میانگین نمرات می‌توان به سطح درس ریاضی کلاس پی برد. مثلاً اگر این میانگین کم باشد باید کلاس تقویتی ریاضی برای دانش‌آموزان آن کلاس برگزار کرد. شکل زیر رابطه بین داده‌ها، عملیات و اطلاعات را از چپ به راست نشان می‌دهد.


شکل ۱ــ۱ــ ارتباط بین داده‌ها، عملیات و اطلاعات

با توجه به شکل ۱ــ۱، مشاهده می‌شود که اطلاعات، حاصل انجام عملیات بر روی داده‌ها است. مجموعه محاسبات و عملیاتی که بر روی داده‌ها صورت می‌گیرد را **پردازش**[2] می‌نامند. پردازش، گاهی ساده مانند محاسبه مجموع و یا خارج قسمت دو عدد می‌باشد و یا گاهی پیچیده مانند تشخیص شماره پلاک خودرو با استفاده از عکس گرفته شده از خودرو، توسط یک کامپیوتر است.

**نکته**

داده‌ها، مقادیر خام و اولیه در مورد یک موضوع هستند.

اطلاعات، نتایج حاصل از عملیات و محاسبات بر روی داده‌ها می‌باشد.

مجموعه محاسبات و عملیاتی که بر روی داده‌ها صورت می‌گیرد را پردازش می‌نامند.

کامپیوتر پردازشگر داده‌ها است.

۱ــ Sensor ۲ــ Process

در فرایند رسیدن از داده‌ها به اطلاعات، نکات زیر باید رعایت گردد :

**۱ــ صحت داده‌ها** : یعنی داده‌ها به درستی گردآوری شده باشند و دادهٔ اشتباه در بین آنها وجود نداشته باشد. مثلاً در بین نمرات درس ریاضی، نمره منفی (کمتر از صفر) و یا بالاتر از بیست نداشته باشیم و همچنین تعداد نمرات با تعداد دانش‌آموزان برابر باشد. و یا در مورد اندازه درجه حرارت محیط، سنسور دما به‌خوبی کار کند و معیوب نباشد و دمای محیط به‌درستی اندازه‌گیری شود.

**۲ــ درستی انجام محاسبات** : یعنی محاسبات و یا به‌طور کلی عملیاتی که بر روی داده‌ها صورت می‌گیرد با دقت و بدون اشتباه انجام شوند و در حین انجام عملیات لطمه‌ای به داده‌ها وارد نشود.

**۳ــ روش انجام پردازش** : با توجه به هدفی که از گردآوری داده‌ها در نظر داریم باید پردازش مناسب نیز بر روی آنها انجام دهیم تا به اطلاعات مفید برسیم. استفاده از روش‌های بهینه و الگوریتم‌های مناسب در عمل پردازش توصیه می‌شود.

معمولاً از کامپیوتر برای انجام پردازش بر روی داده‌ها استفاده می‌کنیم چون سرعت کامپیوتر در اجرای عملیات و دقت انجام محاسبات بالا است. همچنین از کامپیوتر برای ذخیره و نگهداری داده‌ها استفاده می‌شود تا بعداً پردازش بر روی آنها صورت گیرد.

## برای مطالعه

در پیرامون ما دستگاه‌های مختلفی قرار دارند که در هر یک از آنها میکروکامپیوتری برای پردازش داده‌ها وجود دارد. مثلاً در یک دستگاه DVD Player استفاده از لیزر و حسگر، علامت‌های روی دی وی دی به‌عنوان داده‌ها خوانده شده و سپس با پردازش بر روی آنها تبدیل به صدا می‌شوند و به بلندگو فرستاده می‌شوند که برای ما قابل شنیدن و با معنی می‌باشد.

در داخل خودروهای امروزی حداقل یک سیستم کامپیوتری وجود دارد که برای کنترل اندازه تحویل سوخت به موتور به‌کار می‌رود. داده‌های ورودی این کامپیوتر توسط حسگرهای مختلفی که برای اندازه‌گیری مقدار اکسیژن در هوا، اندازه دمای هوا و اندازه سرعت موتور و غیره تعبیه شده‌اند تأمین می‌شود و کامپیوتر پس از پردازش داده‌های دریافتی و مقایسه آنها با داده‌های ثابت موجود در حافظه فقط خواندنی (EPROM)، می‌تواند اندازه سوخت و ترکیب آن با هوا را کنترل نماید و از طریق تزریق سوخت (انژکتور) به موتور بفرستد. سیستم کامپیوتری مربوطه را ECU (Engine Control Unit) می‌نامند. تصور کنید که اگر سنسورها دچار اشکال شوند چه اتفاقی می‌افتد!

## کنجکاوی

وسایلی که در منزل شما قرار دارد و فکر می‌کنید که احیاناً در آنها یک سیستم کامپیوتری کوچک قرار دارد را شناسایی کنید و تشخیص دهید که داده‌های ورودی آن دستگاه چیست و خروجی آن دستگاه کدام است. نتایج تحقیق را به کلاس ارایه دهید.

برای اینکه کامپیوتر بداند که چه پردازشی و یا چه عملیاتی را باید بر روی داده‌ها انجام دهد لازم است دستورات مناسب به آن داده شود.

**به‌مجموعه دستوراتی که به کامپیوتر می‌فهماند که چه نوع پردازشی را بر روی داده‌ها انجام دهد و همچنین اطلاعات به‌دست آمده را چگونه نمایش دهد برنامه[1] می‌گویند.**

دستورات برنامه باید با یک زبان قابل فهم برای کامپیوتر، نوشته شود و با توجه به اینکه سخت‌افزار کامپیوتر بر منطق رقمی (صفر و یک) بنا شده است لذا زبان قابل فهم کامپیوتر دنباله‌ای از کدهای صفر و یک است که به آن زبان **ماشین**[2] می‌گویند. مثلاً برای جمع دو عدد ۵ و ۸ دستورهای زبان ماشین در یک پردازنده شرکت اینتل[3] چنین است :

10111000 00000101 00000000 10111011 00001000 00000000 00000001 11011000

با توجه به اینکه نوشتن دستورات به زبان ماشین وقت‌گیر و دشوار است و زبان ماشین هر پردازنده با پردازنده دیگر متفاوت است، زبان‌های دیگری طراحی شده‌اند که نوشتن برنامه به آن زبان‌ها برای ما ساده‌تر از زبان ماشین است. البته پس از اینکه برنامه با زبان دیگری غیر از زبان ماشین نوشته شد، با استفاده از یک مترجم[4]، باید به زبان ماشین تبدیل شود تا کامپیوتر بتواند آن را بفهمد و اجرا نماید. مترجم خود نیز یک برنامه کامپیوتری می‌باشد که وظیفهٔ آن، ترجمه و تبدیل دستورات یک زبان سطح بالا، به کدهای زبان ماشین می‌باشد.

نوشتن دستورات لازم برای کنترل نحوه کار کامپیوتر، به‌طوری که کامپیوتر بتواند یک کار مشخص را انجام دهد را **برنامه‌نویسی** می‌گویند.

**برنامه‌نویس** شخصی است که آشنا به دستورات یک زبان برنامه‌نویسی باشد و با به‌کارگیری صحیح و مناسب دستورات، برنامه‌نویسی کند.

---

۱ــ Program

۲ــ Mach ne Language

۳ــ Inte

۴ــ Comp er

**نکته**

به مجموعه دستوراتی که به کامپیوتر می‌فهماند که چه نوع پردازشی را باید بر روی داده‌ها انجام دهد و اطلاعات به‌دست آمده را چگونه نمایش دهد برنامه می‌گویند.

زبان قابل فهم سخت افزار کامپیوتر، زبان ماشین نام دارد و متشکل از دنباله‌ای از کدهای ۰ و ۱ است.

برنامه‌ای که با زبانی غیر از زبان ماشین نوشته شود ابتدا باید توسط یک نرم‌افزار مترجم تبدیل به زبان ماشین شود و سپس برنامه ترجمه شده توسط کامپیوتر اجرا می‌گردد.

مترجم خود یک برنامه کامپیوتری است که وظیفه آن ترجمه برنامه نوشته شده به یک زبان، به کدهای زبان ماشین است.

## ۲ــ۱ــ انواع زبان‌های برنامه‌نویسی

زبان‌های برنامه‌نویسی را از نظر این‌که تا چه اندازه به زبان محاوره‌ای ما نزدیک باشند به‌صورت زیر دسته‌بندی می‌کنند :

● **زبان‌های سطح پایین**[۱] : زبان‌هایی که به زبان پردازشگر کامپیوتر (CPU) نزدیک باشد و مسلماً از زبان محاوره‌ای ما دور هستند زبان‌های سطح پایین نام دارند. زبان ماشین و زبان اسمبلی[۲] در گروه زبان‌های سطح پایین قرار دارند.

● **زبان‌های سطح بالا**[۳] : زبان برنامه‌نویسی که به زبان محاوره‌ای ما نزدیک باشد، زبان سطح بالا نام دارد. تاکنون صدها زبان برنامه‌نویسی سطح بالا در دانشگاه‌ها و یا شرکت‌های کامپیوتری طراحی و ابداع شده‌اند ولی بعضی از آنها عمومی نشدند و مورد استقبال برنامه‌نویسان قرار نگرفتند و بعضی از زبان‌ها با آمدن زبان‌های جدید منسوخ شده‌اند. از میان زبان‌های رایج برنامه‌نویسی می‌توان به زبان VB، Java و C# اشاره نمود.

● **زبان‌های سطح میانی**[۴] : زبان‌هایی در این دسته قرار می‌گیرند که در آنها دستوراتی برای دسترسی راحت تر به سخت افزار پیش بینی شده باشد و همچنین به زبان عامیانه نزدیک باشند، مانند زبان C. برنامه‌نویسان از این زبان‌ها برای کار با سخت افزار کامپیوتر و برنامه‌ریزی وسایلی که در آنها

---

۱ــ Low Leve Language
۲ــ Assemb y Language
۳ــ H gh Leve Language
۴ــ Med um Leve Language

پردازشگر وجود دارد استفاده می‌کنند.

در این کتاب با یکی از زبان‌های برنامه‌نویسی به نام C# (بخوانید C Sharp) آشنا خواهید شد که یک زبان سطح بالا می‌باشد.

قبل از شروع به یادگیری زبان برنامه‌نویسی، ممکن است این سؤال طرح شود که با توجه به وجود نرم افزارهای کاربردی[۱] آماده نظیر Office، آیا لزومی‌دارد که یک زبان برنامه‌نویسی را یاد بگیریم تایک نرم افزار جدید بسازیم؟

**پاسخ** : نرم‌افزارهای کاربردی آماده، برای تسهیل و انجام امور عادی و روزمره که بین کاربران مشترک است طراحی شده اند و به تدریج و در طی سالیان طولانی با توجه به بازخورد[۲] کاربران تکمیل تر شده اند. مثلاً برنامه صفحه گسترده[۳] (ابداع شده در سال ۱۹۷۸) یک نرم افزار کاربردی است که افراد و شرکت‌های بسیار زیادی از آن برای انجام امور روزمره خود استفاده می‌کنند. کاربران این گونه نرم افزارها باید از قابلیت‌هایی که در آن نرم افزار از قبل تدارک دیده شده است استفاده کنند و نیاز خود را با استفاده از آن امکانات مرتفع نمایند. با یادگیری یک زبان برنامه‌نویسی این انعطاف را خواهیم داشت که یک نرم افزار کاربردی مطابق با نیاز خودطراحی و برنامه‌نویسی کنیم. همانطور که ممکن است به جای اینکه یک لباس آماده از فروشگاهی خریداری کنیم به سراغ یک خیاط برویم تا یک لباس کاملا سفارشی[۴] و مطابق با سلیقه و مدل مورد نظرما درست کند.

البته طراحی و تولید یک نرم افزار کاربردی، یک کار سنگین، پیچیده و مستلزم صرف وقت زیادی است و به وسیله ی یک گروه از مهندسان با تخصص‌های مختلف طی چند مرحله انجام می‌شود که برنامه‌نویسی تنها یکی از مراحل آن است. البته نگران نباشید ما در این کتاب برنامه‌های کوچک و مقدماتی برای یادگیری زبان برنامه‌نویسی C# می‌نویسیم و در سال بعد شما با مراحل تولید یک نرم افزار و روش‌های طراحی و حل مسئله[۵] آشنا خواهید شد تا بتوانید با استفاده از ابزار برنامه‌نویسی، برنامه‌های کاربردی برای حل یک مشکل در یک سازمان بنویسید و آن را تجربه کنید.

---

۱ــ App cat on Software

۲ــ Feedback

۳ــ Spreadsheet

۴ــ Custom zed

۵ــ Prob em So v ng


**شکل ۲ــ۱ــ نظر یک برنامه‌نویس**

## خودآزمایی فصل اوّل

۱ــ در هر یک از موارد زیر داده، اطلاعات و پردازش را مشخص نمایید.

الف) نرم افزار تهیه کارنامه در مدارس آموزش و پرورش، جمع نمرات و معدل هر دانش آموز را براساس نمراتش، محاسبه و چاپ می کند.

ب) در یک شرکت مهندسی، حقوق دریافتی کارکنان براساس میزان حضورساعات کار در شرکت محاسبه و پس از کسر مالیات و حق بیمه پرداخت می‌شود و بر روی فیش چاپ می‌شود.

ج) یک روش الکترونیکی در پرداخت عوارض خودرو در هنگام عبور از بزرگراه‌ها، استفاده از تکنولوژی RFID[1] است. در این روش یک برچسب[2] مخصوص در روی شیشه خودرو چسبانده می‌شود که دارای یک کد منحصر به فرد است و اطلاعات شماره پلاک خودرو و کد مربوطه و همچنین میزان اعتبار پولی در هنگام خرید برچسب، در کامپیوتر ثبت می‌شود. هر گاه خودرو بدون توقف از محل پرداخت عوارض عبور کند از طریق حس گرهای امواج رادیویی، کد برچسب خودرو شناسایی شده و این کد به کامپیوتر داده می‌شود (شکل ۳ــ۱) و در نتیجه شماره پلاک خودرو و میزان اعتبار آن استخراج می‌شود، عوارض عبور از بزرگراه از اعتبار مربوطه به صورت خودکار کسر می‌شود (شکل ۴ــ۱).

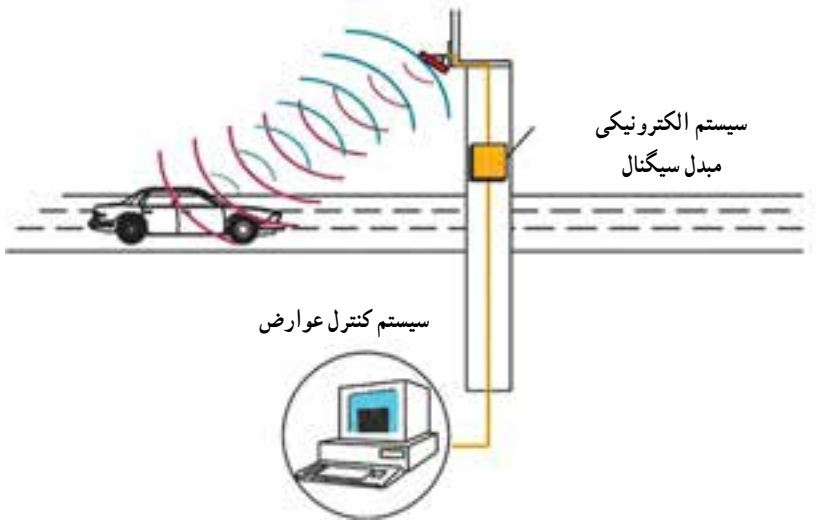
۲ــ من چه سطحی از زبان برنامه‌نویسی هستم؟


شکل ۳ــ۱ــ مرحله اول ورود خودرو به گذرگاه عوارض بزرگراه

---

۱ــ Rad o Frequency Ident f cat on شناسایی با امواج رادیویی

۲ــ Tag




**شکل ۴ــ۱ــ مرحله دوم تشخیص اطلاعات خودرو و کسر عوارض از اعتبار پولی**

تا کنون زبان‌های زیادی در سطح من تولید شده است، اما تنها برخی از آنها مورد اقبال متخصصان قرار گرفته است. من به زبان محاوره‌ای خیلی نزدیک هستم.

۳ــ توضیحات ستون چپ را با اصطلاح ستون سمت راست تطابق دهید. (یک مورد اضافی است.)

| | |
|---|---|
| ۱ــ برنامه | الف ــ برنامه‌ای برای تبدیل کد سطح بالا به کد قابل فهم برای سخت‌افزار |
| ۲ــ برنامه‌نویسی | ب ــ دنباله‌ای از کدهای صفر و یک |
| ۳ــ مترجم | ج ــ مجموعه‌ای از دستورالعمل‌ها برای پردازش داده‌ها |
| ۴ــ زبان ماشین | د ــ نوشتن دستورات لازم برای حل یک مسئله |
| ۵ــ زبان سطح بالا | هـ ــ زبان برنامه‌نویسی نزدیک به زبان محاوره‌ای |
| | وــ داده‌ها و ورودی‌های یک برنامه |

۴ــ در یک برنامه، منظور از ورودی همان ..... و منظور از خروجی همان ..... است.

۵ــ در محاسبهٔ عبارت $z\ x^2\ y$، ورودی (ها) و خروجی (ها) کدامند؟

۶ــ زمان ورود و خروج کارکنان یک شرکت با عکس‌برداری از چهره آنان و به کمک کامپیوتر صورت می‌گیرد. در این روش ورودی (داده)، پردازش و خروجی (اطلاعات) را تعیین کنید.

۷ــ چرا لازم است یک زبان برنامه‌نویسی را یاد بگیریم؟

## فعالیت

۱ــ با استفاده از جستجو در اینترنت، چند زبان سطح بالای متداول امروزی را پیدا کرده و آنها را نام ببرید.

۲ــ دستگاه‌های الکترونیکی و یا کامپیوتری که در زندگی ما وارد شده‌اند و همه به آنها وابسته شده‌ایم مانند تلفن همراه و یا کنسول‌های بازی را در نظر بگیرید. با توجه به کاربرد آنها، داده‌های ورودی به آنها و همچنین با توجه به نوع پردازش، اطلاعات خروجی در آنها را شناسایی کنید.

۳ــ اگر به اتفاق پدر یا مادر به تعمیرگاه اتومبیل رفتید، فرصت را از دست ندهید و از تعمیرکاران سؤال کنید که دستگاه ECU (توضیح داده شده در قسمت «آیا می‌دانید») در کجای خودرو قرار دارد؟ (در همه خودروهای امروزی حتی مانند پراید نیز وجود دارد. تعجب نکنید.) ECU و حس گرهای آن را شناسایی کنید. همچنین وقتی به منزل برگشتید با استفاده از اینترنت منظور از کلمات دیاگ و فلش‌کردن که توسط تعمیرکاران استفاده می‌شود را جستجو کرده و متوجه مفهوم آن شوید.

## واژگان و اصطلاحات انگلیسی فصل اول

| ردیف | واژه انگلیسی | معنی واژه به فارسی |
|---|---|---|
| ۱ | Application Software | |
| ۲ | Assembly Language | |
| ۳ | Compiler | |
| ۴ | Customized | |
| ۵ | Data | |
| ۶ | Feedback | |
| ۷ | High Level Language | |
| ۸ | Information | |
| ۹ | Installation | |
| ۱ | Low Level Language | |
| ۱۱ | Machine Language | |
| ۱۲ | Medium Level Language | |
| ۱۳ | Problem Solving | |
| ۱۴ | Process | |
| ۱۵ | Program | |
| ۱۶ | Sensor | |
| ۱۷ | Spreadsheet | |

# فصل دوم

# آشنایی با زبان #C

در این فصل ابتدا مختصری در مورد پیدایش زبان #C و ارتباط آن با زبان‌های دیگر بیان می‌شود و سپس ساختار یک برنامه به زبان #C معرفی می‌گردد و با دو اصطلاح کلاس و متد به صورت مقدماتی آشنا می‌شوید. در ادامه این فصل روش نام گذاری پاسکال برای انتخاب نام کلاس توضیح داده می‌شود و در انتهای فصل در قسمت کار در کارگاه، اولین برنامه به زبان #C را نوشته و ترجمه و اجرای آن را تجربه خواهید کرد.

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

۱ــ قالب کلی یک برنامه ساده در زبان #C را بیان کند.

۲ــ نحوه تعریف یک کلاس و متد را در یک برنامه ساده بکار بندد.

۳ــ با استفاده از یک ویرایشگر، برنامه ی ساده بنویسد و آن را ذخیره نماید.

۴ــ برنامه ذخیره شده را با استفاده از مترجم در پنجره کنسول ترجمه کرده و سپس اجرا نماید.

۵ــ با استفاده از متدهای مربوط به رنگ، تغییری در خروجی برنامه ی خود به وجود آورد.

## ۱ــ۲ــ آشنایی با زبان #C

زبان #C یک زبان سطح بالا، شی‌گرا[۱] و همه منظوره است که به‌وسیله شرکت مایکروسافت هم زمان با پیدایش لایه نرم‌افزاری جدید آن به نام .NET ابداع و توسعه پیدا کرده است. از نرم‌افزارهای متنوع و گوناگونی از جمله نرم‌افزارهای اداری و برنامه‌های کاربردی تحت وب گرفته تا نرم‌افزارهایی برای تلفن همراه و بازی‌های کامپیوتری، با زبان #C و با استفاده از لایه .NET تولید می‌شود.

زبان #C شباهت زیادی به زبان‌های C و Java دارد و ویژگی‌هایی را از آنها تقلید کرده، یا بعضی

---

۱ــ Object Or ented Language

امکانات آنها را بهبود داده است. تلاش شده است که بهترین ویژگی‌ها گردآوری شود، اما بر خلاف زبان جاوا که متن باز[1] است، #C در انحصار و اختیار سازنده آن یعنی شرکت مایکروسافت است. زبان‌های C و Java هر دو به زبان C برمی‌گردند که در سال ۱۹۷۰ ابداع شد و معروفیت آن به دلیل نوشتن سیستم عامل UNIX به‌وسیله آن بود. زبان C، یک زبان حرفه‌ای است که دست برنامه‌نویس را برای نوشتن برنامه و دسترسی به سخت افزار، باز می‌گذارد و دارای انعطاف بسیار زیادی است، به همین دلیل **کمتر** اشتباهات منطقی برنامه‌نویس را کنترل می‌نماید. اما در زبان #C، در هنگام ترجمه و همچنین اجرای برنامه دقت زیادی بر روی تطبیق و به‌کارگیری داده‌ها صورت می‌گیرد تا از اشتباهات دستوری برنامه‌نویس یا کاربر جلوگیری نماید.

## برای مطالعه

**DENNIS M. RITCHIE**

دنیس ریچی متولد ۱۹۴۱ زبان C را بر مبنای زبان B، طی سال‌های ۱۹۶۹ تا ۱۹۷۳ طراحی کرد و همچنین با آقای Ken Thompson که طراح زبان B بود در طراحی و ایجاد سیستم عامل یونیکس همکاری کرد. وی در سال ۲۰۱۱ در اثر بیماری فوت کرد.

**BJARNE STROUSTRUP**

بیارنه استراس تروپ، زبان C را بر مبنای C و با اضافه کردن ویژگی شی گرایی در سال ۱۹۷۹ طراحی کرد و در سال ۱۹۸۳ این زبان به نام C منتشر شد.

**JAMES ARTHUR GOSLING**

جیمز آرتگاسلینگ متولد سال ۱۹۵۵، به عنو ن پدر زبان جاو شناخته می‌شود. وی در سال ۱۹۹۴ به همر ه تیمش زبان جاو ر بر پایه زبان C و C بد ع کرد وی در بتد نام Oak که نام درختی در جلوی دفتر کار و بود، بر ی ین زبان نتخاب کرد، ما بعدها به Java تغییر نام پید کرد. زبان جاو با هدف ساده کردن و بالابردن منیت و با شعار برنامه ر یک بار بنویس و هر جا جر کن طر حی شده ست.

**ANDERS HEJLSBERG**

اندرس هجلزبرگ مهندس نرم افزارمتولد سال ۱۹۶۰ است وی در ساخت و ابداع چندین زبان برنامه نویسی معروف و موفق همکاری داشته است.او سازنده اصلی زبان توربوپاسکال و دلفی بود و همچنین مدیر تیم طراحی زبان #C است.

**شکل ۱ــ۲ــ پدید آورندگان زبان‌های برنامه‌نویسی C#, Java, C++, C**

---

۱ــ Open Source

## برای مطالعه

### لایه نرم‌افزاری NET Framework. شرکت مایکروسافت

این لایه نرم‌افزاری دارای نسخه‌های مختلفی می‌باشد و همراه با نسخه‌های ویندوز منتشر می‌شود؛ آخرین نسخه آن را می‌توانید از سایت شرکت مایکروسافت دانلود کنید. در حال حاضر (فروردین ۱۳۹۳) نسخه 4.5 ارائه شده است. این لایه شامل موارد زیر است :

- زبان‌های برنامه‌نویسی NET. مانند C#، J# ، F# و VB
- محیطی برای اجرای برنامه‌ها (CLR[1])
- یک سری ابزارهای برنامه‌نویسی مانند مترجم csc که برنامه‌های C# را به کدهای زبان ماشین ترجمه می‌کند.
- یک سری کتابخانه‌های استاندارد مثل ADO.NET که برای دسترسی به بانک های اطلاعاتی می‌باشد.

زبان برنامه‌نویسی C# یک زبان برنامه‌نویسی سطح بالا، همه منظوره و شیء گرا است که به‌وسیلهٔ شرکت مایکروسافت تولید شده است. تیم طراحی این شرکت به سرپرستی آقای Anders Hejlsberg در طی تولید NET Framework. زبان C# را ابداع کرده است. نسخه های مختلف این زبان همراه با NET. عرضه شده است . نسخه C# 5.0 در ۱۵ آگوست ۲۰۱۲ منتشر شده است.

## ۲ــ۲ــ شروع برنامه‌نویسی

همان‌طور که برای تهیه و پخت یک غذا، به مواد اولیه، لوازم آشپزی و دستور پخت[2] نیاز داریم برای تولید یک برنامه نیز، به یک کامپیوتر یا لپ تاپ، لوازم برنامه‌نویسی (یک ویرایشگر متنی[3] و یک برنامه مترجم) و همچنین به یک الگوریتم نیاز داریم. اگر یک کامپیوتر با سیستم عامل ویندوز ۷ یا بالاتر در دسترس باشد تقریباً تمام مواد اولیه و لوازم مورد نیاز را در اختیار داریم.

---

۱ــ Common Language Runt me

۲ــ Rec pe

۳ــ Text Ed tor

در این صورت با طی مراحل زیر می‌توانیم برنامه‌ای را نوشته، ترجمه کرده و سپس اجرا نماییم.

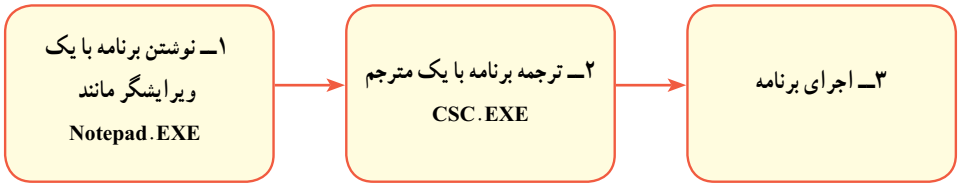
۱ــ نوشتن برنامه و ذخیره آن با استفاده از یک ویرایشگرمانند برنامه Notepad ویندوز

۲ــ ترجمه برنامه ذخیره شده به وسیله مترجم زبان C# به نام CSC.EXE

این مترجم با نصب .NET Framework در روی کامپیوتر قرار می‌گیرد (در پیوست ۲، نحوه نصب آن توضیح داده شده است).

۳ــ اجرای برنامه ترجمه شده

نمودار ۱ــ۲ این مراحل را به ترتیب از چپ به راست نشان می‌دهد.




نمودار ۱ــ۲ــ مراحل برنامه‌نویسی، ترجمه و اجرا

در انتهای این فصل، در قسمت کار در کارگاه، تمام مراحل بالا را به‌صورت عملی تجربه خواهید کرد.

## ۳ــ۲ــ اولین برنامه به زبان C#

با یک برنامه ساده به زبان C# آشنا می‌شویم. برنامه ۱ــ۲ را در زیرمشاهده کنید.

```
class WelcomeToCSharp
{
static void Main( )
   {
         System.Console.WriteLine("Welcome To C#!");
   }
}
```

برنامه ۱ــ۲ــ اولین برنامه به زبان C#

این برنامه کوچک فقط یک پیام خوشامدگویی بر روی صفحه نمایش نشان می‌دهد. رنگ‌های کلمات که در این برنامه مشاهده می‌کنید، تنها برای کمک به واضح شدن برنامه برای خواننده به کار

گرفته شده است و تاثیری بر روی صفحه نمایش ندارد. همان طور که در برنامه Notepad آنچه که می‌نویسید همگی با یک رنگ نوشته می‌شود.

**سؤال:** آیا با مشاهده این برنامه و بدون اطلاعات قبلی و یا کمک از دیگران، می‌توانید حدس بزنید که چه پیامی بر روی صفحه نشان داده می‌شود؟

برای این‌که با این برنامه آشنا شویم و یاد بگیریم که چگونه باید به زبان C# برنامه بنویسیم از دو جنبه این برنامه را بررسی خواهیم کرد :

الف) نگاه جزیی‌تر در حد کلمات و علامت‌ها

ب) نگاه کلی تر در حد تقسیم بندی یک برنامه به قسمت‌های مختلف

با **نگاهی** جزیی تربه برنامه ۱ــ۲، مشاهده می کنید که این برنامه از تعدادی کلمه و علامت تشکیل شده است. بعضی از کلمات مانند static،class،void کلمات شناخته شده برای زبان C# هستند و دارای معنی و مفهوم ثابتی هستند به این نوع کلمات ، کلمات کلیدی یا رزرو شده گفته می‌شود. کلمات رزرو شده[1] به رنگ آبی در این کتاب نوشته شده‌اند و به تدریج با آنها آشنا می‌شوید.

بعضی از کلمات دیگر مانند WelcomeToCSharp نامی‌است که به‌وسیله برنامه‌نویس و طبق سلیقه وی انتخاب می‌شود. به این نام‌ها شناسه[2] می‌گویند. برنامه نویس در انتخاب شناسه‌ها باید ضوابطی را رعایت کند که در فصل چهارم با آن آشنا می‌شوید.

علامت‌هایی مانند }، {، )، ( و نیز در این برنامه دیده می‌شود که معمولاً برای شروع یا پایان یک قسمت استفاده می گردد.

با **نگاهی دیگر و کلی‌تر** به برنامه ۱ــ۲، مشاهده می کنیم که یک برنامه ساده از یک قسمت کلی به نام کلاس تشکیل شده است که با کلمه کلیدی class مشخص می‌شود و شروع و پایان آن با علامت آکولاد باز و بسته تعیین می گردد. در جلوی کلمه کلیدی class، یک نام (شناسه) دلخواه مثلاً WelcomeToCSharp نوشته می‌شود که بیان کننده کار برنامه است. قسمت کلاس برنامه را در شکل زیر مشاهده کنید.

```
class WelcomeToCSharp
{


}
```

کلاس برنامه ۱ــ۲

---

۱ــ Reserved words ۲ــ Ident f er

اگر درون کلاس WelcomeToCSharp را نگاه کنیم یک قسمت دیگر را خواهیم دید که چنین شروع شده است :

```
static void Main( )
```

شروع و پایان این قسمت نیز با علامت‌های آکولاد باز و بسته، مشخص شده است. به این قسمت **متد[1] Main می‌گوییم که بدنه اجرایی برنامه است هر دستوری که در این قسمت نوشته شود به وسیله ی کامپیوتر به ترتیب اجرا می‌شود . دستورهای برنامه خود را در این قسمت می‌نویسیم .**

```
static void Main()
{
System.Console.WriteLine("Welcome To C#!");
}
```

**متد Main برنامه ۱ــ۲**

آخرین قسمتی که در برنامه ۱ــ۲، در داخل متد Main قابل تشخیص است، یک دستور اجرایی است و به کامپیوتر اعلام می‌کند که چه باید انجام دهد که در این برنامه ، نمایش یک پیام است :

```
System.Console.WriteLine("Welcome To C#!");
```

با اجرای دستوربالا، پیام خوشامدگویی Welcome to C#! بر روی صفحه نمایش، نشان داده می‌شود. آیا شما قبلا درست حدس زده بودید که برنامه ۱ــ۲، چه پیامی‌را بر روی صفحه نمایش، نشان می‌دهد؟!

به وسیلهٔ دستور بالا، هر آنچه که داخل علامت‌های نقل قول " " قرار داشته باشد، بر روی صفحه نمایش نشان داده می‌شود حتی اگر به زبانی غیر از انگلیسی مثلا فارسی نوشته شده باشد. توجه داشته باشید که خود علامت های نقل قول بر روی صفحه نمایش، نشان داده نمی‌شوند. بلکه این علامت ها برای مشخص کردن شروع و پایان عبارتی است که می‌خواهیم روی صفحه نشان داده شود.

---

۱ــ Method

**نکته**

توجه داشته باشید که زبان C# مانند زبان‌های C، C++ و Java نسبت به حروف کوچک و بزرگ حساس است و چنانچه قصد دارید برنامه‌ای را در کامپیوتر وارد کنید به دیکته و نوع حروف کوچک و بزرگ کلمات توجه داشته باشید. مثلاً کلمات static و void باید با حروف کوچک نوشته شود ولی حرف اول کلمه Main باید حرف بزرگ (M) باشد.

## ۴ــ۲ــ الگوی یک برنامه ساده به زبان C#

یک برنامهٔ کاربردی نوشته شده به زبان C#، شامل مجموعه‌ای از کلاس‌ها است که هر یک از آنها نیز شامل تعدادی متد هستند. اما در یک برنامه ساده مانند برنامه ۱ــ۲، تنها یک کلاس وجود دارد که در آن نیز فقط یک متد به نام Main( ) تعریف می‌شود که نقطه آغاز اجرای برنامه است و الگوریتم خود را با رعایت قوانین زبان C# در آن می‌نویسیم. الگو یا ساختارکلی یک برنامه ساده به زبان C# در زیر آمده است. الگوی زیر را به‌خاطر بسپارید.

```
class یک نام دلخواه
{
        static void Main()
        {
                دستورات مربوط به انجام یک کار
        }
}
```

**الگوی یک برنامه ساده به زبان C#**

## ۵ــ۲ــ کلاس (class) چیست؟

کلاس یک مفهوم اساسی در برنامه‌نویسی شی گرا است که در کتاب برنامه سازی۲ به تفصیل بحث می‌شود. در اینجا اگر بخواهیم به طور ساده در مورد معنی و مفهوم کلاس صحبت کنیم، باید بگوییم که **کلاس** به عنوان یک قالب یا الگویی می‌باشد که در آن داده‌هایی تعریف می‌شود. این داده‌ها

مربوط به یک موضوع است و عملیاتی که می‌توان بر روی آنها انجام داد. در زبان C# گنجینه‌ای از کلاس‌های مختلف و کاربردی، از قبل تعریف شده و آماده وجود دارد که برنامه نویس کافی است آنها را بشناسد و در برنامه استفاده نماید. Console یک کلاس آماده در زبان C# است که عملیات مختلف ورودی و یا خروجی (بر روی صفحه نمایش و یا صفحه کلید) در آن تعریف شده است.

**۱ــ۵ــ۲ــ نحوه تعریف کلاس :** در زبان C# این امکان برای برنامه نویس فراهم است که کلاس جدیدی را تعریف کند. همان طور که در زیرمشاهده می کنید از کلمه کلیدی class برای تعریف و مشخص کردن یک کلاس جدید استفاده می‌شود. در جلوی کلمه class، یک نام دلخواه ذکر می گردد که نام کلاس است. مانند WelcomeToCSharp

```
class نام کلاس
{
        تعریف داده‌ها
        و عملیات بر روی آنها
}
```

الگوی یک کلاس

**نکته**

**نام گذاری کلاس :** نام یک کلاس به وسیلهٔ برنامه نویس نام گذاری می‌شود. سعی کنید یک نام با معنی و مطابق با کار برنامه انتخاب کنید. ممکن است این نام از چند کلمه تشکیل شده باشد. بین کلمات نباید فاصله بگذارید ولی برای این که خواندنِ نام به راحتی انجام شود و تشخیص کلمات آسان باشد، از **روش پاسکال** استفاده کنید که در آن، اولین حرف هر کلمه با حرف بزرگ نوشته می‌شود.

## ۶ــ۲ــ متد چیست؟

همان‌طور که گفته شد در داخل کلاس، عملیات بر روی داده‌ها و یا الگوریتم انجام یک کار تعریف می‌شود. متد مجموعه‌ای از دستورات است که برای انجام یک کار لازم است. هر متد مطابق با عملکردش نام گذاری می‌شود و همچنین دارای یک جفت پرانتز باز و بسته است که در آن ممکن است ورودی‌هایی ذکر شود که برای انجام کار لازم است.

در برنامه‌های زبان C#، ممکن است متدهای زیادی تعریف و یا مورد استفاده قرارگیرند، اما حتما باید متدی به نام Main() تعریف شده باشد که نقطه آغاز اجرای برنامه است و اجرای یک برنامه از اولین دستور داخل آن شروع می‌شود.

کلمات static و void در قالب کلی متد Main() ویژگی‌های متد را مشخص می‌کنند که در کتاب برنامه سازی ۲ با آن آشنا می‌شوید.

```
static void Main()
{
        دستور شماره ۱ ;
        دستور شماره ۲ ;
        دستور شماره ۳ ;

        ادامه دستورات
}
```

**قالب کلی متد Main()**

در این کتاب مانند برنامه ۱ــ۲، فقط به تعریف متد Main() می‌پردازیم و از متدهای آماده در زبان C# استفاده خواهیم کرد.

**۱ــ۶ــ۲ــ استفاده از متدهای آماده :** تعداد زیادی متد در کلاس‌های آماده زبان C# مانند وجود دارد که هر یک از آنها، برای انجام کاری در نظر گرفته شده است. مثلا متد WriteLine() از کلاس Console برای نشان دادن پیام روی صفحه نمایش در نظر گرفته شده است که در برنامه ۱ــ۲ از آن استفاده کردیم :

```
System.Console.WriteLine("Welcome To C#!");
```

همان‌طور که برای آدرس دادن منزل خود به دیگران، نام منطقه، خیابان، کوچه و شماره پلاک را ذکر می‌کنید، برای استفاده از یک متد نیز باید نام فضا یا حوزه، نام کلاس و سپس نام متد را مشخص کنید و برای جدا کردن آنها از یکدیگر، علامت نقطه بین آنها بنویسید (نمودار ۲ــ۲).


نمودار ۲ــ۲ــ طریقه استفاده از یک متد (System.Conso e.Wr teL ne)

به این ترتیب برای استفاده از متد ()WriteLine در برنامه ۱ــ۲ مشاهده می‌کنید که فضای نامی System[1] و نام کلاس Console و در آخر نام متد نوشته شده است که با علامت نقطه از یکدیگر جدا شده‌اند. متدهای دیگری نیز در کلاس Console وجود دارد که در این فصل با برخی از آنها آشنا می‌شوید.

برنامه زیر برای نمایش دو پیام بر روی صفحه نوشته شده است :

```
class WelcomeToCSharp
{
static void Main()
   {
         System.Console.WriteLine("Welcome To C#!");
         // Insert a blank line
         System.Console.WriteLine();
         System.Console.WriteLine("This is my first program.");
   }
}
```

برنامه ۲ــ۲ــ استفاده از توضیحات در متن برنامه

برنامه ۲ــ۲ مانند برنامه ۱ــ۲ است با این تفاوت که سه خط دیگر به متد ()Main اضافه شده است.

خط اول فقط یک توضیح[2] برای خواننده برنامه و یا برنامه نویسان می‌باشد و توضیح می‌دهد که خط بعدیِ برنامه چه عملی را انجام می‌دهد. نشانه توضیحات علامت // (دو بار کلید /) است و مترجم با دیدن این علامت متوجه می‌شود که این خط یک توضیح است؛ بنابراین آن را به زبان ماشین ترجمه نمی‌کند.

---

۱ــ NameSpace

۲ــ Comment

خط دوم یک دستور اجرایی است :

```
System.Console.WriteLine();
```

در این خط از متد ( ) WriteLine استفاده شده است با این تفاوت که داخل پرانتز، خالی است. اجرای این دستور سبب می‌شود که روی صفحه نمایش یک سطر خالی ایجاد شود. دستور آخر، پیام .This is my first program را روی صفحه نمایش نشان می‌دهد.

با توجه به برنامه ۲ـ۲، پیامی را که می‌خواهید نمایش داده شود باید بین علامت های نقل قول " " قرار دهید. برای مثال اگر اسم شما Mohammad است و بخواهید روی صفحه نشان داده شود، باید به‌صورت زیر بنویسید :

```
System.Console.WriteLine ("Mohammad");
```

اگر بخواهید نام و نام خانوادگی خود را در دو سطر، نمایش دهید، در اینصورت می‌توانید دو بار از متد WriteLine() استفاده کنید. مثلا برای نمایش "Mohammad " و " Ghasemi" چنین می‌نویسیم :

```
System.Console.WriteLine("Mohammad");
System.Console.WriteLine("Ghasemi");
```

حروف، علامت‌ها و عبارتی که ما بین علامت‌های نقل قول نوشته می‌شود را **رشته**[1] می‌نامند. این حروف می‌تواند فارسی، انگلیسی یا به هر زبانی باشد. مانند نام "Mohammad" و یا یک رمز عبور "Mehran2014". هر یک از حروف و علامت‌ها را نیز یک **کاراکتر**[2] می‌نامند. برای مثال Mohammad از ۸ کاراکتر و رمز عبور Mehran2014 از ۱۰ کاراکتر تشکیل شده است. چنان‌چه فاصله در رشته وجود داشته باشد، فاصله نیز یک کاراکتر محسوب می‌شود.

**نکته**

برای درج توضیحات در برنامه، اگر یک خط باشد از علامت // و چنان‌چه چند خط باشد از علامت /* توضیحات */ استفاده کنید :

```
// Display a greeting message
/*
    FileName: welcome.cs ... Date : 05_07_2014
    Display a greeting message
*/
```

۱ـ Str ng ۲ـ Character