۱ 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:

- مفهوم داده، اطلاعات و پردازش را بیان کند و آنها را به کار بندد.
- تعریف برنامه، برنامه‌نویسی، مترجم و انواع زبان‌های برنامه‌نویسی را از نظر سطح نزدیکی به زبان محاوره‌ای بیان کند.
- در یک برنامه ورودی، خروجی و پردازش را معین کند.
- 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 با 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.

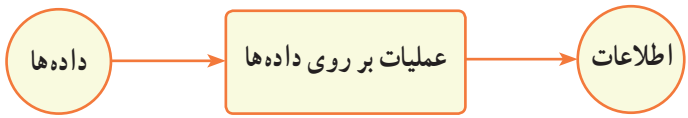
۱۷/۵ , ۱۴ , ۱۹ , ۱۱ , ۱۸ , ...

۱ــ Data & information

اسامی دانش‌آموزان یک کلاس و یا نام شهرهای استان محل سکونت شما مثال‌های دیگری از داده‌ها 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 با 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 ارائه 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 Machine Language
۳ــ Intel
۴ــ Compiler

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 Level Language
۲ــ Assembly Language
۳ــ High Level Language
۴ــ Medium Level 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[1] آماده نظیر Office، آیا لزومی‌دارد که یک زبان برنامه‌نویسی را یاد بگیریم تا یک نرم افزار جدید بسازیم؟

پاسخ: نرم‌افزارهای کاربردی آماده، برای تسهیل و انجام امور عادی و روزمره که بین کاربران مشترک است طراحی شده‌اند و به تدریج و در طی سالیان طولانی با توجه به بازخورد[2] کاربران تکمیل‌تر شده‌اند. مثلاً برنامه صفحه گسترده[3] (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 خود طراحی 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 کاملاً سفارشی[4] 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 وسیلهٔ 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 نرم‌افزار و روش‌های طراحی و حل مسئله[5] 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 Application Software

۲ــ Feedback

۳ــ Spreadsheet

۴ــ Customized

۵ــ Problem Solving

شکل ۲ــ۱ــ نظر یک برنامه‌نویس

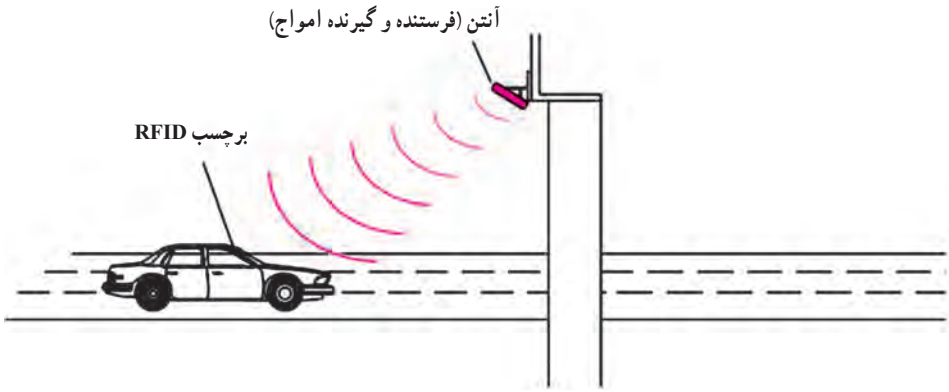
خودآزمایی فصل اول

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.

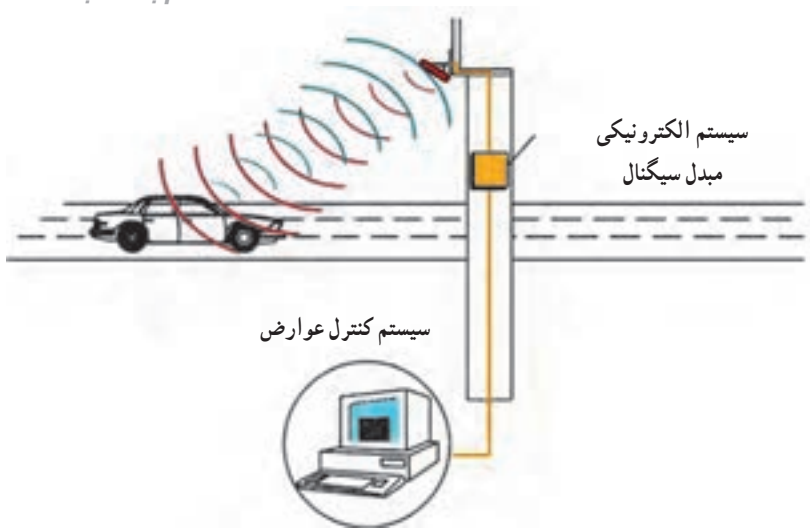
ج) یک روش الکترونیکی در پرداخت عوارض خودرو در هنگام عبور از بزرگراه‌ها، استفاده از تکنولوژی RFID[1] است. در این روش یک برچسب[2] مخصوص در روی شیشه خودرو چسبانده می‌شود که دارای یک کد منحصر به فرد است و اطلاعات شماره پلاک خودرو و کد مربوطه و همچنین میزان اعتبار پولی در هنگام خرید برچسب، در کامپیوتر ثبت 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).



شکل ۳ــ۱ــ مرحله اول ورود خودرو به گذرگاه عوارض بزرگراه

۱ــ Radio_Frequency Identification شناسایی با امواج رادیویی

۲ــ Tag



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

۲ــ حدس بزنید چه سطحی از زبان برنامه‌نویسی هستم؟

تاکنون 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 اوّل

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	
۱۰	Low Level Language	
۱۱	Machine Language	
۱۲	Medium Level Language	
۱۳	Problem Solving	
۱۴	Process	
۱۵	Program	
۱۶	Sensor	
۱۷	Spreadsheet	

۲ 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:

- قالب کلی یک برنامه ساده در زبان C# را بیان کند.
- نحوه تعریف یک کلاس و متد را در یک برنامه ساده بکار بندد.
- با استفاده از یک ویرایشگر، برنامهٔ ساده بنویسد و آن را ذخیره نماید.
- برنامه ذخیره شده را با استفاده از مترجم در پنجره کنسول ترجمه کرده و سپس اجرا نماید.
- با استفاده از متدهای مربوط به رنگ، تغییری در خروجی برنامهٔ خود به وجود آورد.

۱ـ۲ـ 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 امکانات آنها را بهبود داده است. تلاش شده است که بهترین ویژگی‌ها گردآوری شود، اما بر خلاف زبان جاوا که متن باز[۲] است، C# در انحصار و اختیار سازنده آن یعنی شرکت مایکروسافت است. زبان‌های C++

۱ـ Object Oriented Language ۲ـ Open Source

و Java هر دو به زبان C برمی‌گردند که در سال ۱۹۷۰ ابداع شد و معروفیت آن به دلیل نوشتن سیستم‌عامل UNIX به‌وسیله آن بود. زبان C، یک زبان حرفه‌ای است که دست برنامه‌نویس را برای نوشتن برنامه و دسترسی به سخت‌افزار، 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 عنوان پدر زبان جاوا شناخته می‌شود. وی در سال ۱۹۹۴ به همراه تیمش زبان جاوا را بر پایه زبان C و C++ ابداع کرد وی در ابتدا نام Oak که نام درختی در جلوی دفتر کار او بود، برای این زبان انتخاب کرد، اما بعدها به Java تغییر نام پیدا کرد. زبان جاوا با هدف ساده کردن و بالابردن امنیت و با شعار برنامه را یک بار بنویس و هر جا اجرا کن طراحی شده است.

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

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 Runtime

۲ــ Recipe

۳ــ Text Editor

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

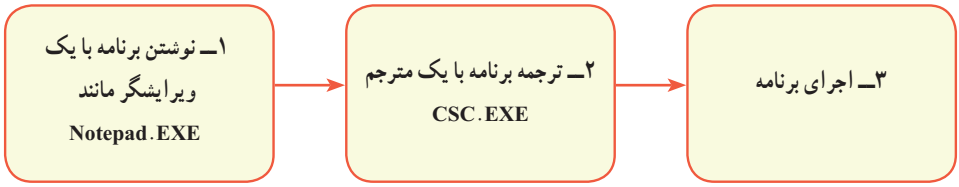
۱ــ نوشتن برنامه و ذخیره آن با استفاده از یک ویرایشگر مانند برنامه Notepad ویندوز

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 خوش‌آمدگویی 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 تأثیری بر روی برنامه 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** جزیی‌تر به برنامه ۱ــ۲، مشاهده می‌کنید که این برنامه از تعدادی کلمه و علامت تشکیل شده است. بعضی از کلمات مانند class،static و 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 نامی است 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 می‌گردد.

با **نگاهی دیگر و کلی‌تر** به برنامه ۱ــ۲، مشاهده می‌کنیم 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 می‌گردد. 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 Identifier

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 وسیلهٔ 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 خوش‌آمدگویی Welcome to C#!بر روی صفحه نمایش، نشان داده می‌شود. آیا شما قبلاً 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 مثلاً فارسی نوشته شده باشد. توجه داشته باشید که خود علامت های نقل قول بر روی صفحه نمایش، نشان داده نمی‌شوند. بلکه این علامت‌ها 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 حتماً 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# وجود دارد که هر یک از آنها، برای انجام کاری در نظر گرفته شده است. مثلاً 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 قرار دهید (نمودار ۲ــ۲).

نمودار ۲ــ۲ــ طریقه استفاده از یک متد (System.Console.WriteLine)

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 دوم این برنامه فقط یک توضیح[2] برای خواننده برنامه 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 سوم 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 ۲ــ۲، 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. مثلاً 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
*/
```

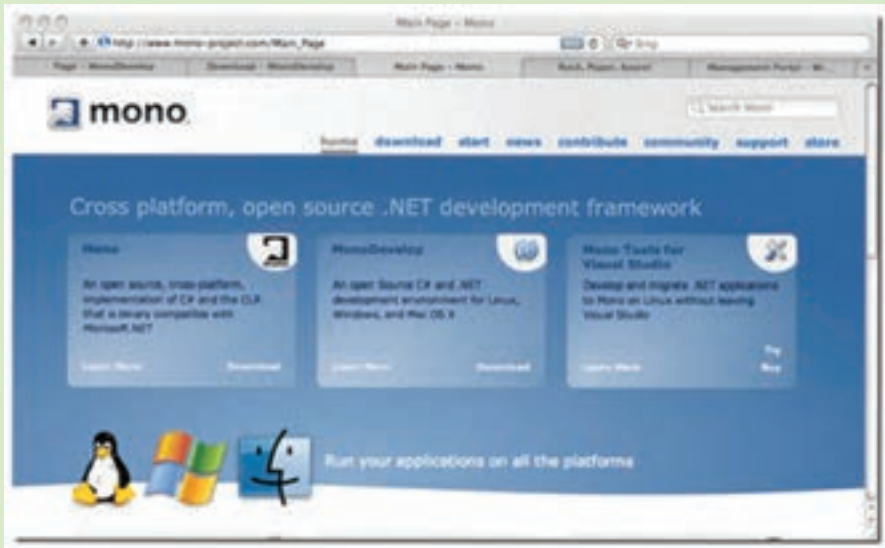
۱ــ String ۲ــ Character

برای مطالعه

زبان برنامه‌نویسی #C از لایه نرم‌افزاری .NET استفاده می‌کند و مایکروسافت .NET را برای ویندوز طراحی کرده و به این ترتیب روی ویندوز کار می‌کند و اگر کسی بخواهد بر روی سیستم عامل دیگری، برنامه #C را اجرا کند باید لایه نرم‌افزاری مطابق با .NET را بر روی آن سیستم داشته باشد. خوشبختانه برنامه‌هایی مانند Mono وجود دارند که بر روی سیستم عامل‌های دیگری غیر از ویندوز نیز نصب می‌شود و دارای مترجم #C می‌باشد و به این وسیله برنامه‌های نوشته شده به این زبان را بر روی سیستم‌های دیگر غیر از ویندوز می‌توان ترجمه و اجرا کرد. برنامه Mono برای سیستم عامل‌های زیر وجود دارد :

Android, BSD, iOS, Linux, OS X, Windows, Solaris and UNIX

سری به سایت Mono بزنید. www.go-mono.com

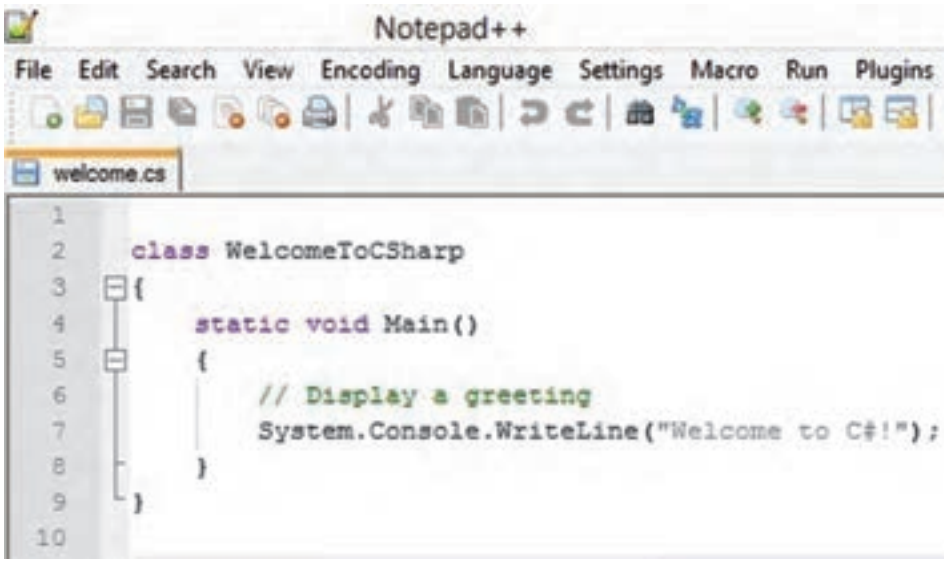


شکل ۲ــ۲ــ برنامه نویسی #C بر روی سیستم عامل غیر از ویندوز

کار در کارگاه: برنامه نویسی به زبان C#

قدم اول : نوشتن و تایپ برنامه : برای نوشتن یک برنامه ساده مانند برنامه ۱ــ۲، که در این فصل مورد بررسی قرار گرفت، نیاز به یک ویرایشگر متنی[۱] است. یک ویرایشگر متنی قادر است حروف، کلمات و آنچه را که تایپ می‌کنید بدون در نظر گرفتن اطلاعات نوع فونت، اندازه حروف و رنگ در یک فایل ذخیره کند. برنامه Notepad یک ویرایشگر ساده است که همراه با سیستم عامل ویندوز در روی کامپیوتر نصب می‌شود.

علاوه بر برنامه Notepad، می‌توانیم از ویرایشگر دیگری مانند برنامه Notepad++ که به‌صورت رایگان از طریق سایت[۲] آن قابل دانلود است استفاده نماییم. این ویرایشگر به‌منظور برنامه‌نویسی به زبان‌های مختلف طراحی شده است به‌طوری که کلمات رزرو شده، رشته‌ها و توضیحات در این ویرایشگر با رنگ های مختلف نشان داده می‌شود (شکل ۳ــ۲). البته برای استفاده از این ویژگی باید ابتدا از منوی Language، زبان برنامه‌نویسی موردنظر خود را C# انتخاب کنید.



شکل ۳ــ۲ــ تصویری از محیط ویرایشگر Notepad++

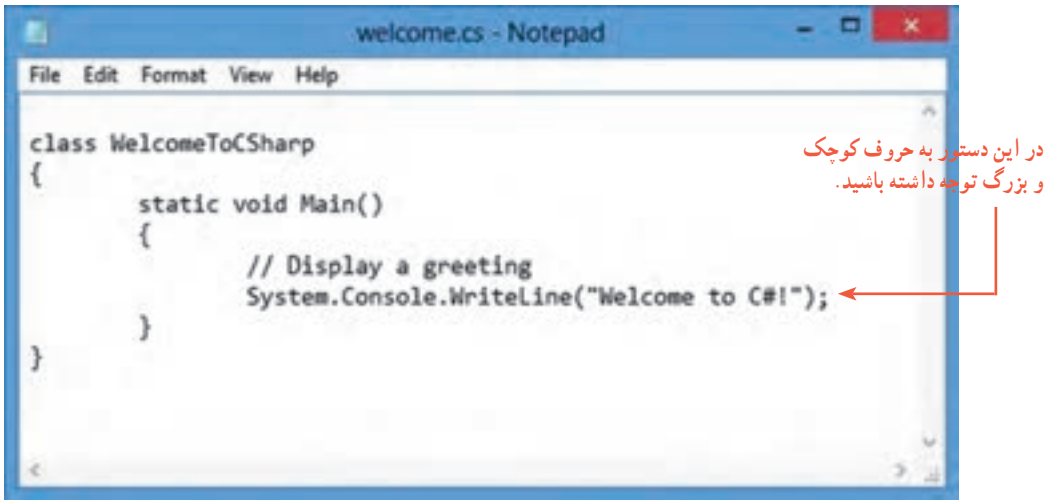
۱ــ Text Editor

۲ــ http: //notepad-plus-plus.org

توجه داشته باشید که از **برنامه Word** استفاده نکنید که در این صورت، کدهای اضافی مربوط به صفحه‌بندی، رنگ و فونت را نیز به فایل شما اضافه می‌کند که مترجم در هنگام ترجمه برنامه، انتظار آنها را ندارد و دچار مشکل می‌شود.

● در هنگام تایپ دستورات، صرفنظر از اینکه از چه ویرایشگری (Notepad , Notepad++) استفاده می‌کنید، باید دقت داشته باشید که زبان C# نسبت به حروف کوچک و بزرگ حساس است؛ بنابراین برنامه را دقیقاً مانند کتاب تایپ کنید (شکل ۴ــ۲).

پس از آنکه برنامه ۱ــ۲ را تایپ کردیم، باید آن را ذخیره نماییم. برای این منظور به منوی File بروید و گزینه Save As... را انتخاب کنید. در هنگام ذخیره فایل دقت کنید که فایل را در کجا و در چه مسیری ذخیره می‌نمایید و نام فایل مناسبی را برای آن انتخاب کنید و پسوند آن را cs. قرار دهید (شاید بهتر باشد یک پوشه در یکی از درایوها به نام خودتان بسازید و فایل را در آن ذخیره کنید). در این تمرین نام فایل را welcome.cs قراردهید.

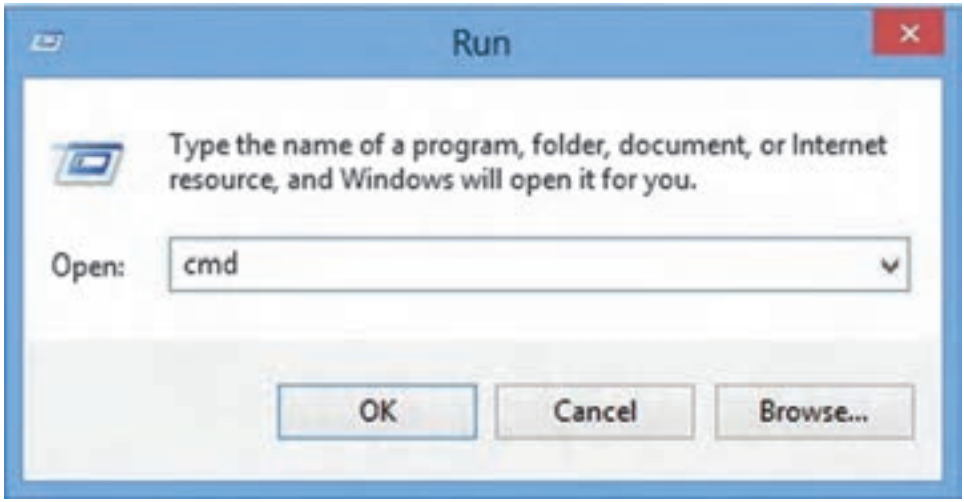


شکل۴ــ۲ــ تصویری از محیط ویرایشگر Notepad

قدم دوم : ترجمه و اجرای برنامه : پس از نوشتن و ذخیره کردن برنامه، لازم است ابتدا برنامه را ترجمه کنیم و اگر اشکالی در تایپ آن وجود دارد آن را برطرف کنیم.

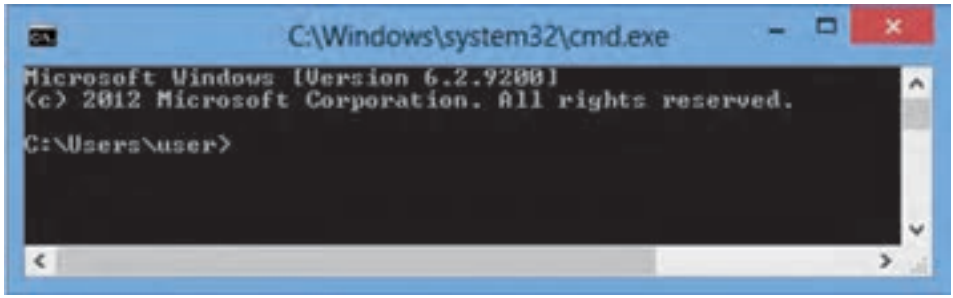
برای ترجمه برنامه مراحل زیر را دنبال کنید.

۱ــ از طریق گزینه Run فرمان cmd را اجرا کنید، تا وارد پنجره فرمان شویم (شکل ۵ــ۲).



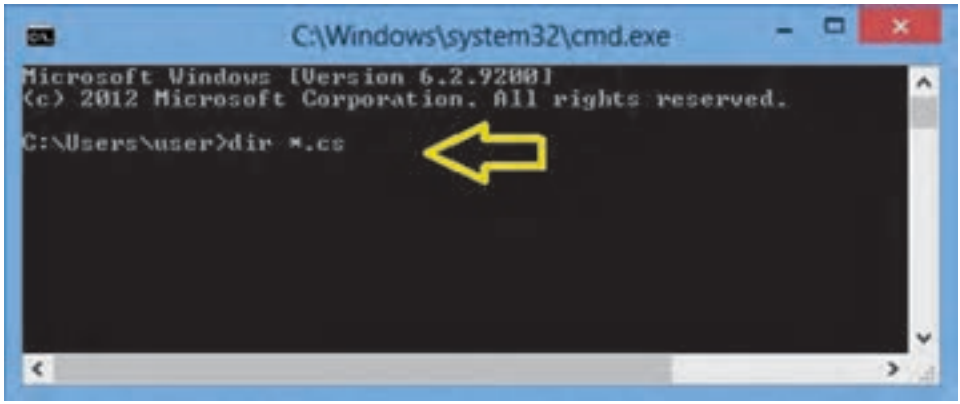
شکل ۵ــ۲ــ فرمان cmd

۲ــ پنجره فرمان ظاهر می‌شود (نوشته‌ها در شکل، ممکن است با آنچه در پنجره command prompt کامپیوتر شما دیده می‌شود متفاوت باشد) (شکل ۶ــ۲).



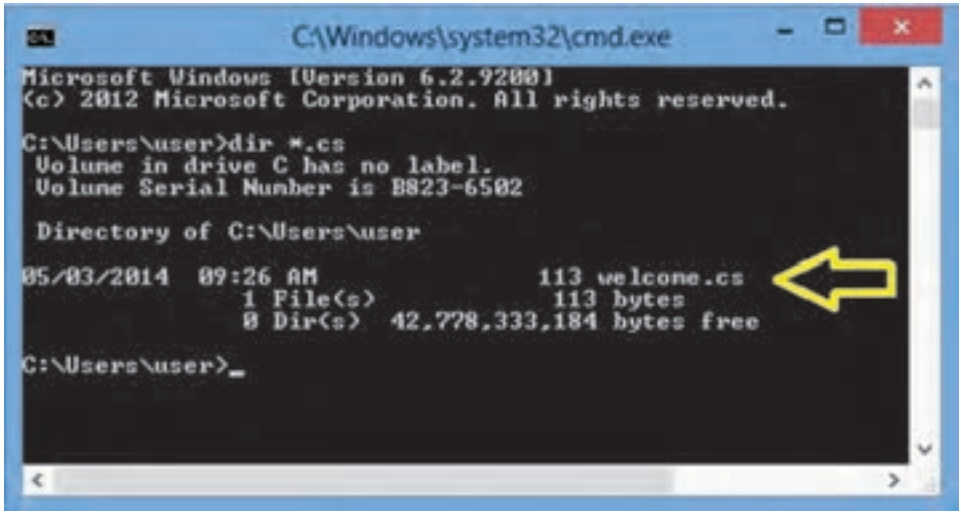
شکل ۶ــ۲ــ پنجره command prompt

۳ــ در پنجره Command Prompt از فرمان Dir استفاده می‌کنیم و با توجه به این‌که پسوند فایل cs می‌باشد، با تایپ فرمان زیر از وجود فایل برنامه مطمئن می‌شویم (شکل ۷ــ۲). اگر فایل را پیدا نکردید باید وارد پوشه‌ای شوید که برنامه را در آنجا ذخیره کرده‌اید. به این ترتیب از دستور cd برای وارد شدن به پوشه موردنظر خود استفاده کنید. شاید دستور cd.. نیز برای وقتی که می‌خواهید به یک پوشه بالاتر بروید مفید باشد.



شکل ۷ــ۲ــ فرمان Dir

۴ــ اگر مرحله قبل را به درستی انجام دهید، باید شکل ۸ــ۲ را مشاهده کنید.

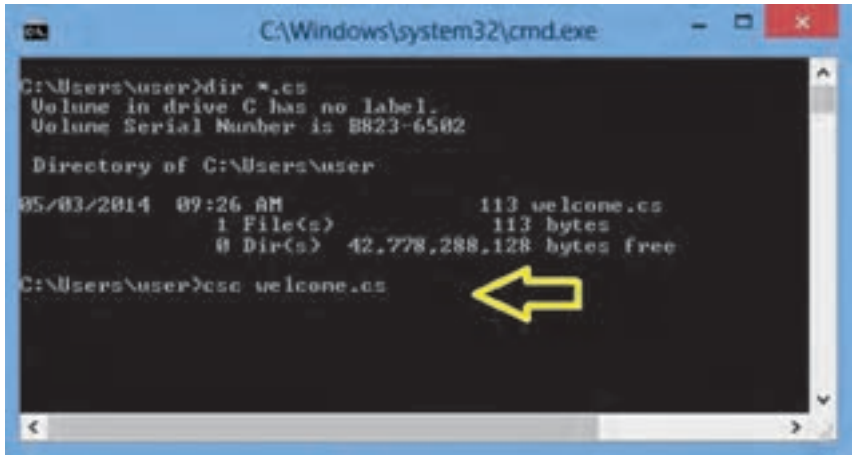


شکل ۸ــ۲ــ مشخصات فایل برنامه

۵ــ توجه داشته باشید اگر برنامه Net Framework. بر روی کامپیوتر شما قبلاً نصب شده باشد برنامه‌ای به نام csc.exe[1] برای ترجمه برنامه‌های زبان C# در اختیار دارید. پس از یافتن فایل خود، با استفاده از این مترجم، برنامه خود را ترجمه نمایید. در پنجره فرمان از دستور زیر استفاده کنید :

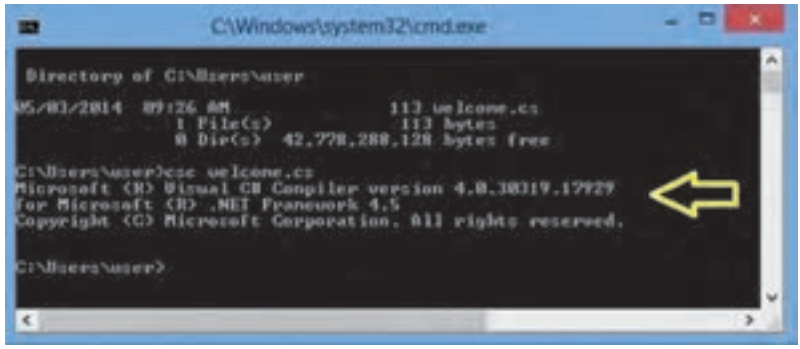
```
csc filename.cs
```

منظور از filename.cs نام فایل مورد نظر شما است.



شکل ۹ــ۲ــ فرمان ترجمه برنامه

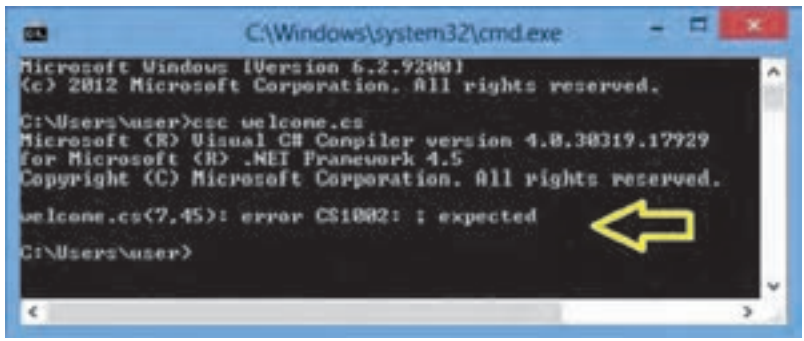
۶ــ با اجرای دستور بالا مترجم شروع به ترجمه برنامه می‌کند و اگر همه کارها به درستی انجام شده باشد (برنامه NET. قبلاً نصب شده باشد، تایپ دستور csc را درست انجام داده باشید، فایل csc به درستی در مسیر جستجوی سیستم عامل (path) معرفی شده باشد و برنامه هیچ خطایی نداشته باشد.) شکل ۱۰ــ۲ را مشاهده خواهید کرد.



شکل ۱۰ــ۲ــ نتیجه ترجمه برنامه

۱ــ C Sharp Compiler

۷ــ اگر در تایپ برنامه اشتباهی انجام داده باشید، مترجم نمی‌تواند برنامه را ترجمه کند و در این‌صورت خطا یا خطاهای برنامه را گزارش می‌دهد. برای مثال اگر فراموش کرده باشید علامت نقطه ویرگول را در انتهای دستور System.Console.WriteLine() بنویسید، با ترجمه برنامه خطای شکل ۷ــ۲ ظاهر می‌شود.



شکل ۱۱ــ۲ــ پیام خطای مترجم برای ننوشتن علامت ; در خط ۷ برنامه

در خط آخر این شکل مشاهده می‌کنید که از سمت چپ، ابتدا نام فایل برنامه یعنی welcome.cs و سپس در جلوی آن دو عدد ۷ و ۴۵ نوشته شده است که مکان خطا را در برنامه نشان می‌دهد. عدد اول (۷) شماره سطر و عدد دوم (۴۵) شماره ستون محل خطا در برنامه است. بعد از این دو عدد، کد خطا (error CS1002) و سپس توضیح آن (; expected) ذکر شده است. شما باید توضیح خطا را با دقت بخوانید و معنی آن را درک کنید تا بتوانید اشکال برنامه را برطرف کنید. در این مثال توضیح خطا چنین است : علامت ; فراموش شده

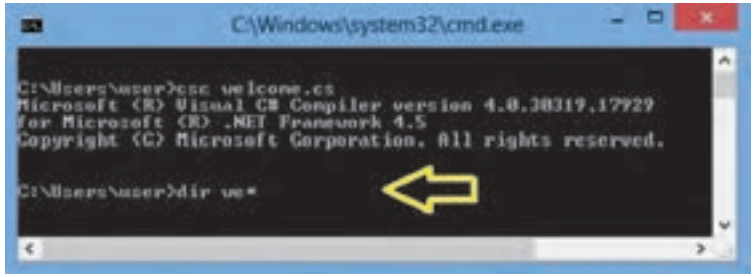
برای برطرف کردن خطای بالا لازم است برنامه را به وسیلهٔ ویرایشگر باز کنید و به سطر ۷ و ستون ۴۵ بروید و علامت نقطه ویرگول را اضافه کنید و سپس برنامه را تحت همان نام قبلی ذخیره کنید و دوباره به پنجره فرمان بازگردید و عمل ترجمه را انجام دهید. اگر خطایی رخ نداد به مرحله بعدی بروید وگرنه باید دوباره عملیات رفع اشکال را تکرار کنید.

نکته

اشتباه معمول برنامه نویسان در هنگام نوشتن برنامه، فراموش کردن علامت نقطه ویرگول در انتهای دستورات است. در این صورت خطای صادره از طرف مترجم چنین است :

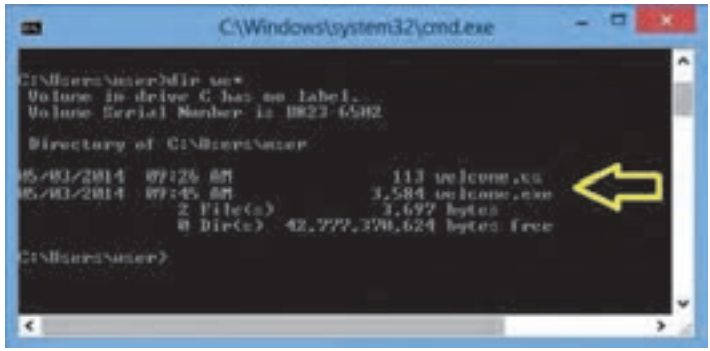
error CS1002: ; expected

۸ــ در صورت ترجمه موفق برنامه، فایل جدیدی ساخته می‌شود. با دستور Dir از وجود آن مطمئن می‌شویم (شکل ۱۲ــ۲).



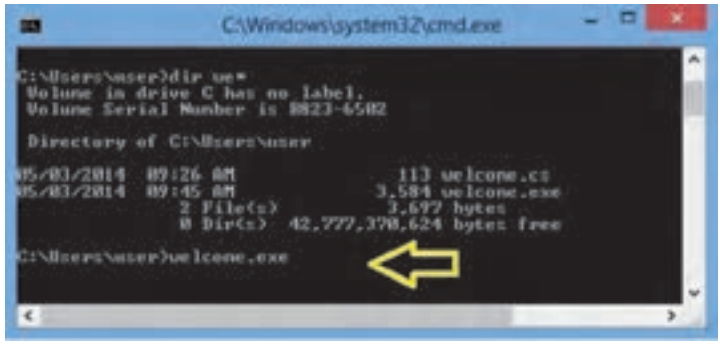
شکل ۱۲ــ۲ــ جستجوی فایل ترجمه شده

۹ــ همان طور که در شکل ۱۳ــ۲ مشاهده می‌کنید، فایل جدیدی با همان نام welcome ولی با پسوند EXE ساخته شده است.



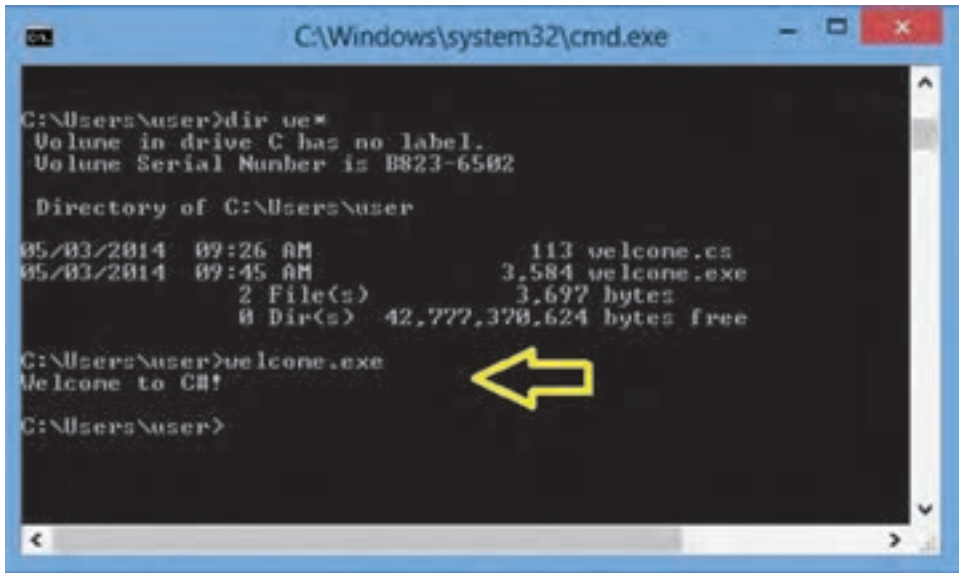
شکل ۱۳ــ۲ــ لیست فایل های برنامه و ترجمه شده

۱۰ــ بعد از اینکه فایل اجرایی ساخته شد، می‌توانید برنامه را اجرا نمایید. کافی است نام آن را در پنجره فرمان بنویسید و کلید Enter را بزنید (شکل ۱۴ــ۲).



شکل۱۴ــ۲ــ اجرای فایل ترجمه شده یا برنامه اجرایی

۱۱ــ نتیجه اجرای برنامه به‌صورت شکل ۱۵ــ۲ خواهد بود.

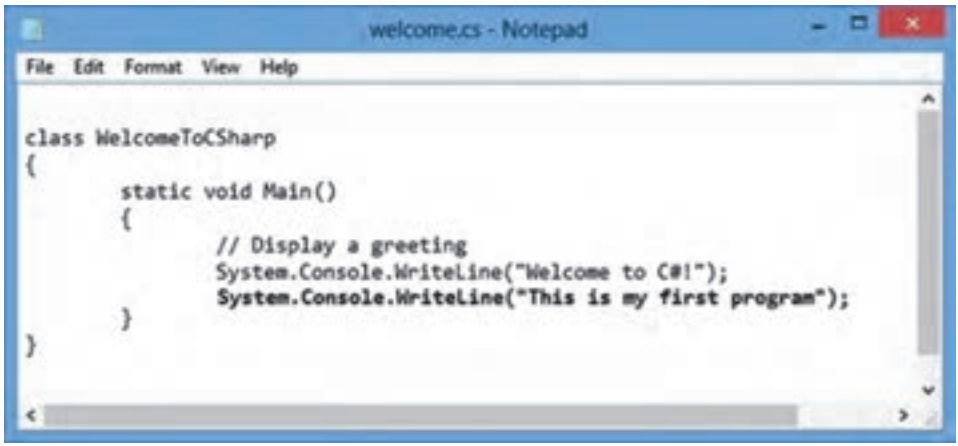


شکل ۱۵ــ۲ــ نتیجه اجرای برنامه

تبریک می‌گوییم که توانستید اولین برنامه خود را به زبان C# بنویسید و آن را ترجمه و اجرا نمایید.

۱۲ــ برنامه را با ویرایشگر باز کنید و یک دستور مانند شکل ۱۶ــ۲ به آن اضافه کنید. (تغییرات با رنگ تیره مشخص شده است.)

برنامه را تحت همان نام قبلی ذخیره کنید و مراحل ترجمه و اجرا (مرحله ۵ به بعد) را تکرار کنید.



شکل ۱۶ــ۲ــ اضافه کردن یک دستور به برنامه

۱۳ــ دوباره برنامه را با ویرایشگر باز کنید و در اولین دستور به جای متد ()WriteLine از متد ()Write مانند شکل ۱۷ــ۲ استفاده کنید.

شکل۱۷ــ۲ــ استفاده از متد ()Write

چه تفاوتی در اجرای برنامه حاصل می‌شود؟ پیام‌ها چگونه نشان داده شدند؟

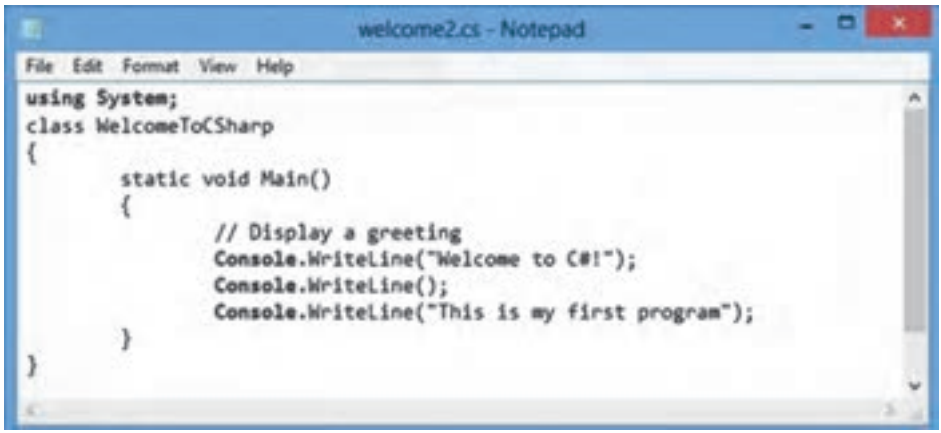
۱۴ــ برنامه را به حالت اولیه خود بازگردانید و یک دستور نیز به‌صورت شکل ۱۸ــ۲ به آن اضافه کنید. برنامه را ذخیره کنید و سپس ترجمه و اجرا نمایید. چه تغییری در خروجی ایجاد می‌شود؟

شکل ۱۸ــ۲ــ استفاده از متد ()WriteLine برای ایجاد یک خط خالی

۱۵ــ همان‌طور که در شکل ۱۸ــ۲ مشاهده می‌کنید، برای هر بار استفاده از متد WriteLine() مجبور هستیم کلمات System و Console را ذکر کنیم. برای کوتاه کردن این دستورات می‌توانیم در ابتدای برنامه، فضای نامی System را معرفی کنیم که در آن کلاس Console تعریف شده است. در این‌صورت می‌توانیم در داخل برنامه، کلمه System را از ابتدای دستورات حذف کنیم. برای معرفی فضای نامی از دستور using به‌صورت زیر استفاده می‌کنیم :

```
using فضای نامی ;
```

برنامه قبلی را باز کرده و تغییرات زیر را اعمال کنید و آن را اجرا نمایید (شکل ۱۹ــ۲).

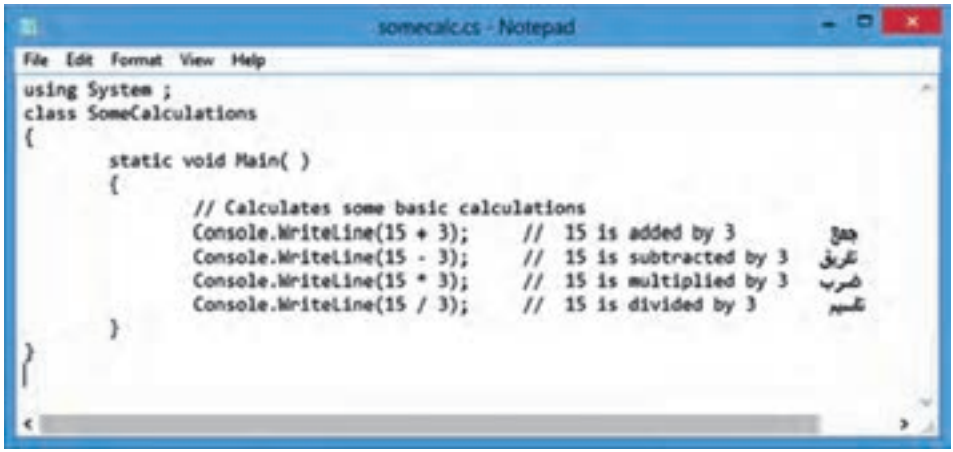


شکل ۱۹ــ۲ــ استفاده از فضای نام

۱۶ــ از متد WriteLine() علاوه بر نمایش یک پیام می‌توان نتیجه یک عبارت ریاضی را نیز نمایش داد. برنامه‌ای برای انجام چهار عمل اصلی بر روی دو عدد نوشته شده است (شکل ۲۰ــ۲). این برنامه را با توجه به نکته زیر تایپ نموده و سپس ترجمه و اجرا نمایید. و به اعداد نشان داده شده برروی صفحه توجه کنید. آیا محاسبات درست انجام شده است؟

نکته

برای این‌که سریع‌تر بتوانید این برنامه را تایپ کنید، یکی از برنامه‌های قبلی خود را با ویرایشگر باز کنید و فقط نام کلاس و دستورات داخل متد Main را تغییر دهید و سپس تحت نام جدیدی آن را ذخیره (Save As...) کنید. توضیحات برنامه می‌تواند به زبان فارسی نیز باشد.



شکل ۲۰ــ۲ــ برنامهٔ چهار عمل اصلی

۱۷ــ برنامه شکل ۲۰ــ۲ را با اعداد دیگری آزمایش کنید یعنی به جای ۳ و ۱۵ اعداد دلخواه دیگری مانند ۲۵ و ۴۰ قرار دهید و سپس برنامه را ترجمه و اجرا نمایید. اعدادی که روی صفحه نشان داده می‌شوند را یادداشت کنید، آیا محاسبات درست انجام شده است؟

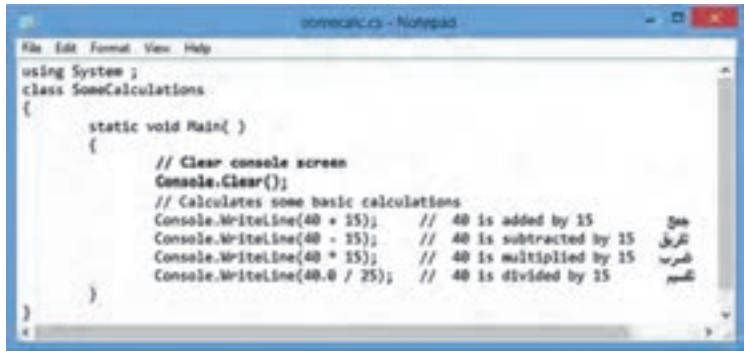
۱۸ــ اگر به خروجیِ دستور زیر توجه کنید

```
Console.WriteLine(40 / 25);
```

مشاهده خواهید کرد که عدد ۱ چاپ می‌شود چون تقسیم صحیح و بدون اعشار انجام می‌شود. در صورتی‌که جواب صحیح و دقیق عدد ۱/۶ است. برای این که کامپیوتر را مجبور به انجام عمل تقسیم اعشاری کنیم، لازم است دست کم یکی از اعداد را به صورت اعشاری بنویسیم. یعنی عدد ۴۰ را به صورت ۴۰/۰ یا عدد ۲۵ را به صورت ۲۵/۰ بنویسید. حال برنامه را در ویرایشگر باز کرده و یکی از اعداد در دستور تقسیم را به‌صورت اعشاری بنویسید و سپس برنامه را با همان نام قبلی ذخیره و ترجمه و اجرا نمایید.

```
Console.WriteLine(40.0 / 25);
```

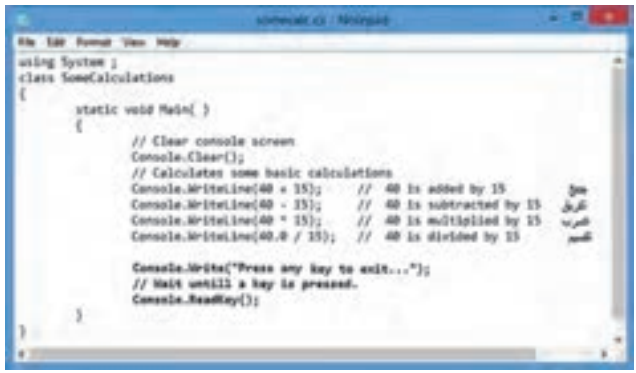
۱۹ــ برای این که پیام‌ها بهتر دیده شوند، می‌توانیم ابتدا با استفاده از متد Clear() صفحه نمایش را پاک کنیم. به این منظور برنامه را با ویرایشگر باز کنید و متد مذکور را به برنامه طبق شکل ۲۱ــ۲ اضافه کنید. برنامه را تحت نام فعلی ذخیره کنید و سپس ترجمه و اجرا نمایید.



شکل ۲۱ــ۲ــ استفاده از متد ()Clear برای پاک کردن صفحه

۲۰ــ پنجره فرمان را ببندید وارد محیط میزکار[1] شوید و با استفاده از Computer وارد پوشه‌ای شوید که فایل‌های خود را در آن ذخیره کرده‌اید. فایل‌های اجرایی (EXE) تولید شده را پیدا کنید. با دوبار کلیک بر روی آیکن آن، برنامه را اجرا کنید. چه اتفاقی می‌افتد؟

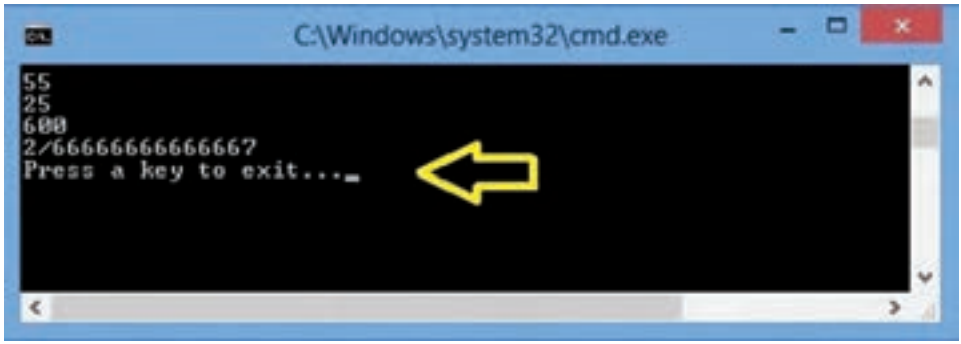
در یک لحظه صفحه کنسول باز شده و به سرعت بسته می‌شود و شما فرصت نمی‌کنید نتایج برنامه را مشاهده کنید. لازم است در انتهای برنامه، دستوری بنویسید که کامپیوتر را به‌صورت موقتی متوقف کند تا شما بتوانید نتایج محاسبات را ببینید. به این منظور از متد ()ReadKey استفاده می‌کنیم. با اجرای این متد، کامپیوتر منتظر زدن کلیدی باقی می‌ماند. البته برای این که کاربر بداند که چرا اجرای برنامه متوقف شده است و کامپیوتر منتظر دریافت چه چیزی است، یک پیام با استفاده از متد ()Write روی صفحه چاپ می‌کنیم. شکل ۲۲ــ۲ استفاده از این متدها را نشان می‌دهد. برنامه چهار عمل اصلی را باز کنید و تغییرات زیر را در آن اعمال و برنامه را ترجمه و اجرا کنید.



شکل ۲۲ــ۲ــ اضافه کردن متد ()ReadKey به برنامه شکل ۲۱ــ۲

۱ــ Desktop

با اجرای برنامه، مشاهده خواهید کرد که نتایج محاسبات و پیام ...Press any key to exit روی صفحه نشان داده می‌شود و سپس کامپیوتر متوقف می‌شود و با زدن یک کلید برنامه پایان می‌یابد (شکل ۲۳ــ۲).



شکل ۲۳ــ۲ــ استفاده از متد ReadKey() برای دریافت یک کلید و توقف موقتی برنامه

کار با رنگ‌ها : ConsoleColor جعبهٔ رنگ ۱۶تایی در C# است. نام این رنگ‌ها در جدول ۱ــ۲ آمده است.

۲۱ــ ویژگی BackgroundColor رنگ زمینه کنسول را نشان می‌دهد.

مثال : با استفاده از جعبه رنگ ConsoleColor رنگ زمینه را آبی کنید :

```
Console.BackgroundColor = ConsoleColor.Blue ;
```

با اجرای دستور بالا تغییری در رنگ زمینه مشاهده نمی‌شود. برای تغییر رنگ زمینه لازم است از متد Clear کلاس Console استفاده کنیم. دستور زیر را در خط بعدی بنویسید.

```
Console.Clear( ) ;
```

۲۲ــ ویژگی ForegroundColor رنگ قلم کنسول را نشان می‌دهد.

مثال : کلمه IRAN را با رنگ قرمز روی صفحه بنویسید :

```
Console.ForegroundColor = ConsoleColor.Red ;
Console.WriteLine("IRAN") ;
```

جدول ۱ــ۲ ــ جعبهٔ رنگ ConsoleColor

نام رنگ در ConsoleColor	نام رنگ	نمونه رنگ
Black	مشکی	Black
DarkBlue	سورمه ای	DarkBlue
DarkGreen	سبز تیره	DarkGreen
DarkCyan	فیروزه ای تیره	DarkCyan
DarkRed	قرمزتیره	DarkRed
DarkMagenta	بنفش	DarkMagenta
DarkYellow	زرد تیره	DarkYellow
DarkGray	خاکستری تیره	DarkGray
Blue	آبی	Blue
Green	سبز	Green
Cyan	فیروزه‌ای	Cyan
Red	قرمز	Red
Magenta	صورتی	Magenta
Yellow	زرد	Yellow
White	سفید	White
Gray	خاکستری	Gray

خودآزمایی فصل دوم

۱ــ قالب کلی یک برنامه به زبان C# چگونه است؟

۲ــ درهر برنامه به زبان C#، دست‌کم یک ..... تعریف می‌شود که در داخل آن، یک متد به نام ..... وجود دارد که اجرای برنامه از آن نقطه آغاز می‌گردد.

۳ــ در زبان C#، انتهای یک دستور با چه علامتی مشخص می‌شود؟

۴ــ تحقیق کنید که در زبان‌های C، C++ و Java علامت پایان دستور چیست؟

۵ــ علامت توضیحات در زبان C# کدام است؟

۶ــ روش قراردادی پاسکال برای نوشتن نام یک کلاس چیست؟

۷ــ برای نشان دادن یک پیام یا یک عبارت بر روی صفحه نمایش از چه دستوری استفاده می‌کنید؟

۸ــ تفاوت بین دو متد ()WriteLine و ()Write را بیان کنید.

۹ــ با استفاده از متد ()WriteLine، دستور مناسبی برای ایجاد یک سطر خالی بین نوشته‌ها در خروجی بنویسید.

۱۰ــ مثالی از کاربرد دستور using بیاورید.

۱۱ــ برای هر یک از خواسته‌های زیر، دستور یا دستورات مربوطه را بنویسید :

الف) نام شما در خروجی بارنگ زرد نمایش داده شود.

ب) صفحه کنسول نمایش را پاک کنید.

پ) اجرای برنامه تا فشردن یک کلید متوقف شود.

ت) صفحه کنسول نمایش با رنگ آبی پاک شود.

تمرینات برنامه‌نویسی فصل دوم

۱ــ برنامه‌ای بنویسید که نام، نام خانوادگی و نام مدرسه شما را روی صفحه نمایش دهد.

۲ــ برنامه تمرین ۱ را تغییری دهید تا اطراف نام شما یک کادر مانند شکل زیر رسم نماید مثلاً اگر اسم شما محمد است. روی صفحه چنین نمایش داده شود :

```
- - - - - - - - - - -
| MOHAMMAD |
- - - - - - - - - - -
```

۳ــ برنامه‌ای بنویسید که حرف انگلیسی نام شما را با استفاده از علامت * نشان دهد. مثلاً اگر اسم شما حمید است حرف انگلیسی H را با استفاده از علامت * به‌صورت زیر نشان دهد.

```
*     *
*     *
******
*     *
*     *
```

یادآوری : هنگامی‌که می‌خواهید برنامه‌ای بنویسید سعی کنید از برنامه‌ای که قبلاً ذخیره کرده‌اید استفاده کنید. برنامه را در یک ویرایشگر باز و سپس تغییرات لازم را ایجاد و ذخیره نمایید.

۴ــ برنامه‌ای بنویسید که تعداد روزهای عمر شما را با استفاده از ضرب سن شما در عدد ۳۶۵ محاسبه روی صفحه نشان دهد. مثلاً اگر سن شما ۱۶ است عبارت ۱۶ × ۳۶۵ را محاسبه و نمایش دهد.

```
System.Console.WriteLine (16 * 365);
```

۵ــ با توجه به تمرین ۴ تعداد سال‌های کبیسه را نیز حساب کنید و برنامه را تغییر دهید.

(**راهنمایی :** برای محاسبه تعداد سال کبیسه، خارج قسمت سن بر عدد ۴ را حساب کنید.)

۶ــ با اجرای برنامه زیر چه عبارتی بر روی صفحه نشان داده می‌شود؟ به نظر شما چه ارتباطی بین علایم {۰} و {۱} و {۲} با اعداد ۱۸ و۱۵ و ۱۵+۱۸ وجود دارد؟

```
class SomeCalculations
{
static void Main( )
   {
   System.Console.WriteLine("{0} + {1} = {2}" , 18 , 15 , 18+15);
   }
}
```

۷ــ تمرین زیر به زبان انگلیسی است. آن را با دقت بخوانید و برنامه خواسته شده را بنویسید. برای زیبایی خروجی از جعبه رنگ ConsoleColor استفاده نمایید.

Write a program that prints a face, using text characters, hopefully better looking than this one.

```
  ///////
 | o o |
(|  ^  |)
 |  —  |
```

واژگان و اصطلاحات انگلیسی فصل دوم

ردیف	واژه انگلیسی	معنی واژه به فارسی
۱	C Sharp Compiler	
۲	Calculate	
۳	Calculation	
۴	Character	
۵	Class	
۶	Command Prompt	
۷	Comment	
۸	Common Language Runtime	
۹	Console	
۱۰	Display a greeting message	
۱۱	Expected	
۱۲	Main	
۱۳	Method	
۱۴	Namespace	
۱۵	Object Oriented Language	
۱۶	Open Source	
۱۷	Pascal case	
۱۸	Path	
۱۹	Press any key to exit	
۲۰	Reserved words	
۲۱	String	
۲۲	Text Editor	