به نام خدا

آموزش زبان

Microsoft C# .NET

جلد اول

مؤلف:

روزبه امیر عصامی

مؤسسه فرهنگی هنری
دیباگران تهران

◂ عنوان کتاب: **آموزش زبان**

Microsoft C# .NET

◂ **مولف : روزبه امیرعصامی**

◂ **ناشر: موسسه فرهنگی هنری دیباگران تهران**

◂**ویراستار:** مهدیه مخبری

◂**صفحه آرایی:** نازنین نصیری

◂**طراح جلد:**

◂ **نوبت چاپ:** اول

◂ **تاریخ نشر:** ۱۴۰۲

◂ **چاپ و صحافی:**

◂ **تیراژ:**۱۰۰ جلد

◂ **قیمت:**

◂ **شابک:** ۹۷۸-۶۲۲-۲۱۸-۷۴۵-۳

نشانی واحد فروش:تهران، خیابان انقلاب،خیابان دانشگاه
-تقاطع شهدای ژاندارمری-پلاک ۱۵۸ساختمان دانشگاه-
طبقه دوم-واحد۴ تلفن ها: ۶۶۹۶۵۷۴۹-۲۲۰۸۵۱۱۱

فروشگاههای اینترنتی دیباگران تهران :
WWW.MFTBOOK.IR
www.dibagarantehran.com

سرشناسه:
عنوان و نام پدیدآور:
مشخصات نشر: تهران : دیباگران تهران :۱۴۰۲
مشخصات ظاهری:۲۸۶ ص:مصور،
شابک: ۹۷۸-۶۲۲-۲۱۸-۷۴۵-۳
وضعیت فهرست نویسی: فیپا
موضوع:
موضوع:
موضوع:
رده بندی کنگره:
رده بندی دیویی:
شماره کتابشناسی ملی:

نشانی تلگرام:mftbook@ **نشانی اینستاگرام دیبا dibagaran_publishing**

فهرست مطالب

مقدمه ناشر ۸

مقدمه ۹

قسمت اول ۱۱

قسمت دوم ۱۷

قسمت سوم ۲۲

Float و Double ۲۲

Decimal ۲۳

دریافت ورودی از کاربر ۲۵

قسمت چهارم ۲۸

کاراکترها (Characters) ۲۸

بولین (The bool Type) ۳۰

قسمت پنجم ۳۳

Program Control Statements ۳۳

The if statement ۳۳

قسمت ششم ۳۸

شرط‌های تودرتو (Nested ifs) ۳۹

The if-else-if Ladder ۴۰

قسمت هفتم ۴۲

عملگرهای افزایشی و کاهشی (Increment Decrement) ۴۲

قسمت هشتم ۴۶

حلقهٔ for (The for loop) ۴۶

حلقهٔ while (The while loop) ۴۹

تمرین ۵۰

قسمت نهم ۵۱

حل تمرین‌های ۱ تا ۴ ..... ۵۴

قسمت دهم ۵۶

حلقهٔ do-while ۵۸

حل تمرین شماره ۵ ..... ۵۹

قسمت یازدهم ۶۲

استفاده از break برای خارج شدن از حلقه ۶۲

استفاده از continue ۶۴

The switch statement ۶۶

قسمت دوازدهم ۶۹

The goto ۷۲

قسمت سیزدهم ۷۶

تمرین ۷۸

قسمت چهاردهم ۸۰

حل تمرین شماره ۶ ..... ۸۰

حل تمرین شماره ۷ ..... ۸۱

حل تمرین شماره ۸ ..... ۸۳

حل تمرین شماره ۹ ..... ۸۴

حل تمرین شماره ۱۱ ..... ۸۶

قسمت پانزدهم ۸۷

قسمت شانزدهم ۹۴

عملگرهای منطقی (Logical Operators) ۹۴

عملگر Implication ۹۵

عملگرهای منطقی Short-Circuit یا اتصال کوتاه ۹۷

عملگرهای بیتی (The Bitwise Operators) ۹۸

آرایه ۱۰۰

آرایه یک‌بعدی ۱۰۰

قسمت هفدهم ۱۰۳

تمرین ۱۱۳

قسمت هجدهم ۱۱۴

آرایه‌های چندبعدی ۱۱۴

آرایه دوبعدی ۱۱۴

آرایه‌های چندبعدی ۱۱۶

آرایه‌های دنده‌دار (Jagged Array) ۱۱۸

Implicitly Typed Variable ۱۲۰

Implicitle Typed Array ۱۲۱

حلقهٔ Foreach ۱۲۲

حل تمرین شماره ۱۲ ..... ۱۲۳

قسمت نوزدهم ۱۲۹

برنامه‌نویسی شیءگرا (Object-Oriented Programming) ۱۲۹

مفاهیم برنامه‌نویسی شیءگرا ۱۳۰

Class چیست؟ ۱۳۰

Object چیست؟ ۱۳۰

چگونه یک Object ساخته می‌شود؟ ۱۳۴

Method چیست؟ ۱۳۵

Return کردن از یک Method ۱۳۷

استفاده از پارامترها ۱۴۱

قسمت بیستم ۱۴۲

Constructor و چگونگی استفاده از آن ۱۴۲

کلمه‌کلیدی this ۱۴۶

String ها در سی‌شارپ ۱۴۸

آرایهٔ string ها ۱۴۹

کلمهٔ کلیدی static ۱۵۱

کلاس static ۱۵۶

قسمت بیست و یکم ۱۵۸

تمرین ۱۶۹

قسمت بیست و دوم ۱۷۱

حل تمرین شماره ۱۳ ..... ۱۷۱

قسمت بیست و سوم ۱۹۳

قسمت بیست و چهارم ۲۰۵

فرستادن Reference به متدها ۲۰۵

استفاده از پارامترهای ref و out ۲۱۰

ref Parameter Modifier ۲۱۰

out Parameter Modifier ۲۱۲

استفاده از argument به تعداد دلخواه ۲۱۵

قسمت بیست و پنجم ۲۱۷

Return کردن object از متد ۲۱۷

Return کردن یک آرایه ۲۱۸

Method Overloading ۲۲۰

Overload Constructors ۲۲۳

درخواست یک overloaded constructor از طریق this ۲۲۶

تمرین ۲۲۹

قسمت بیست و ششم ۲۳۰

هنگامی‌که یک متغیر تعریف می‌کنید، دقیقاً چه اتفاقی می‌افتد؟ ۲۳۰

Value types و Reference types ۲۳۲

Boxing and Unboxing ۲۳۴

Garbage Collection ۲۳۵

Object Initializers ۲۳۵

Optional Arguments ۲۳۶

Named Arguments ۲۳۸

قسمت بیست و هفتم ۲۴۰

حل تمرین شماره ۱۴ ..... ۲۴۰

قسمت بیست و هشتم ۲۵۱

کلاس Tune ۲۵۱

کلاس Album ۲۵۲
کلاس Artist ۲۵۳
کلاس MusicBox ۲۵۵
کلاس UI ۲۵۵
کلاس Program ۲۵۷
قسمت بیست و نهم ۲۶۲
قسمت سی‌ام ۲۷۰
Operator Overloading ۲۷۰
اصول Operator Overloading ۲۷۰
ادامهٔ حل تمرین شماره ۱۴ ..... ۲۷۴
قسمت سی و یکم ۲۷۹

مقدمه ناشر

خط مشی انتشارات مؤسسه فرهنگی هنری دیباگران تهران در عرصه کتاب هایی با کیفیت عالی است که بتواند خواسته های به روز جامعه فرهنگی و علمی کشور را تا حد امکان پوشش دهد.

هر کتاب دیباگران تهران، یک فرصت جدید شغلی و علمی

حمد و سپاس ایزد منان را که با الطاف بیکران خود این توفیق را به ما ارزانی داشت تا بتوانیم در راه ارتقای دانش عمومی و فرهنگی این مرز و بوم در زمینه چاپ و نشر کتب علمی و آموزشی گام‌هایی هرچند کوچک برداشته و در انجام رسالتی که بر عهده داریم، مؤثر واقع شویم.

گستردگی علوم وسرعت توسعه روزافزون آن، شرایطی را به وجود آورده که هر روز شاهد تحولات اساسی چشمگیری در سطح جهان هستیم. این گسترش و توسعه، نیاز به منابع مختلف از جمله کتاب را به عنوان قدیمی‌ترین و راحت‌ترین راه دستیابی به اطلاعات و اطلاع‌رسانی، بیش از پیش برجسته نموده است.

در این راستا، واحد انتشارات مؤسسه فرهنگی هنری دیباگران تهران با همکاری اساتید، مؤلفان، مترجمان، متخصصان، پژوهشگران و محققان در زمینه های گوناگون و مورد نیاز جامعه تلاش نموده برای رفع کمبودها و نیازهای موجود، منابعی پُربار، معتبر و با کیفیت مناسب در اختیار علاقمندان قرار دهد.

کتابی که در دست دارید تألیف **"جناب آقای روزبه امیرعصامی"** است که با تلاش همکاران ما در نشر دیباگران تهران منتشرگشته و شایسته است از یکایک این گرامیان تشکر و قدردانی کنیم.

با نظرات خود مشوق و راهنمای ما باشید

با ارائه نظرات و پیشنهادات وخواسته های خود،به ما کمک کنید تا بهتر و دقیق تر در جهت رفع نیازهای علمی و آموزشی کشورمان قدم برداریم.برای رساندن پیام هایتان به ما از رسانه های دیباگران تهران شامل سایتهای فروشگاهی و صفحه اینستاگرام و شماره های تماس که در صفحه شناسنامه کتاب آمده استفاده نمایید.

مدیر انتشارات

مؤسسه فرهنگی هنری دیباگران تهران
dibagaran@mftplus.com

مقدمه

برنامه‌نویسان همیشه بـه دنبـال راهـی بـرای بهترکـردن کـارایی، سـودمندی و قابـل حمـل بـودن (اسـتفاده در سیستم‌های عامل مختلف) برنامه‌های خود می‌گردند. بدین منظور همیشه دنبال ابزارهای زیادی هستند که از آنها استفاده کنند. زبان‌های برنامه‌نویسی زیادی در دنیا وجود دارند، اما تعداد اندکی از آنها معتبر و عالی هسـتند. یـک زبان برنامه‌نویسی عالی باید قدرتمند و درعین‌حال انعطاف‌پذیر، نحو و گرامر (syntax) آن باید مختصر و مفید و درعین‌حال واضح و روشن باشد.

شرکت مایکروسافت در ژوئن سـال ۲۰۰۰ پلتفـرم Netو زبـان برنامـه‌نویسـی C# (سـی شـارپ) را بـه دنیـای برنامه‌نویسی ارائه داد. مایکروسافت در مصاف با جاوا به دنبال ارائه یک زبان کامل بود که حضور جاوا را در ایـن میدان خیلی کم‌رنگ‌تر کند که تا امروز به هدف خود بسیار نزدیک شده اسـت و توانسـته گسـتردگی و مقبـولیتی به‌مراتب بیشتر از جاوا نزد توسعه‌دهندگان نرم‌افزار پیدا کند.

سی شارپ مستقیماً از دو زبان خیلی موفق C و++C گرفته شده است. نحـو و گرامـر (syntax) و بسـیاری از کلمات کلیدی و عملگرهای آن از زبان C و مدل شیءگرایی آن از++C تأثیر پذیرفته، ایـن زبـان همچنـین بـه شدت از زبان‌های جاوا و دلفی مشتق شده است.

در طول تاریخ کامپیوتر، زبان‌های برنامه‌نویسی تکامل یافتند تا با تغییرات در زمینهٔ کامپیوتر و همهٔ تفکرات نـوین در مورد برنامه‌نویسی منطبق شوند، همان‌طور که همهٔ برنامه‌نویسان می‌دانند هیچ‌چیز به مدت طولانی در دنیـای برنامه‌نویسی ثابت باقی نمی‌ماند، سی شارپ هم از این قاعده مستثنا نبود و توانایی بالای خودش را در پاسخ‌گویی سریع به نیازهای برنامه‌نویسان نشان داد. از سال ۲۰۰۰ که اولین نسخهٔ سی شارپ (C# 1.0) منتشر شد تا بـه امروز که آخرین نسخه از سی شارپ (C# 4.0) منتشر شده ویژگی‌های بسیاری به آن افـزوده شـده اسـت و در این مقالات سعی بر این است که از جدیدترین ویژگی‌ها در آموزش و مثال‌ها استفاده شود.

سی شارپ زبان برتر شرکت مایکروسافت و مادر زبان‌های برنامه‌نویسی Net.است. برخلاف بـاور بعضـی‌هـا کـه تصور می‌کنند سی شارپ تنها برای محصولات خود شرکت مایکروسافت ازجملـه ویندوز، ویندوز موبایـل و... است، سی شارپ به‌گونه‌ای طراحی شده که وابستگی به یک پلتفرم خاص را ندارد. شما با زبـان C# مـی‌توانیـد برای سیستم‌های عامل ویندوز، مکینتاش، ویندوز موبایل، آندروید، iOS، لینوکس و... نرم‌افزار تولید کنید کـافی است یک سر به سایتmono بزنید و اطلاعات کامل در این مورد را بدست آورید.

بدون شک تسلط و فراگیری زبان سی شارپ به منزله یک پتانسیل با ارزش بوده کـه ثمـرات آن بـرای شـما در آینده بیشتر هویدا خواهد شد. استاندارد بودن و وجود کتابخانه‌های مملو از کلاس‌های مفید به شما این اطمینان را می‌دهد که با یادگیری این زبان به یک توانایی فرامحیطی جدید دست پیدا خواهید کرد و بتوانید در سیستم‌هـای عامل مختلف نرم‌افزار خود را تولید کنید و از برنامه‌نویسی لذت ببرید. اگر برای برنامه‌نویسی زبان سـی شـارپ را انتخاب کرده‌اید، انتخاب شما کاملاً صحیح است.

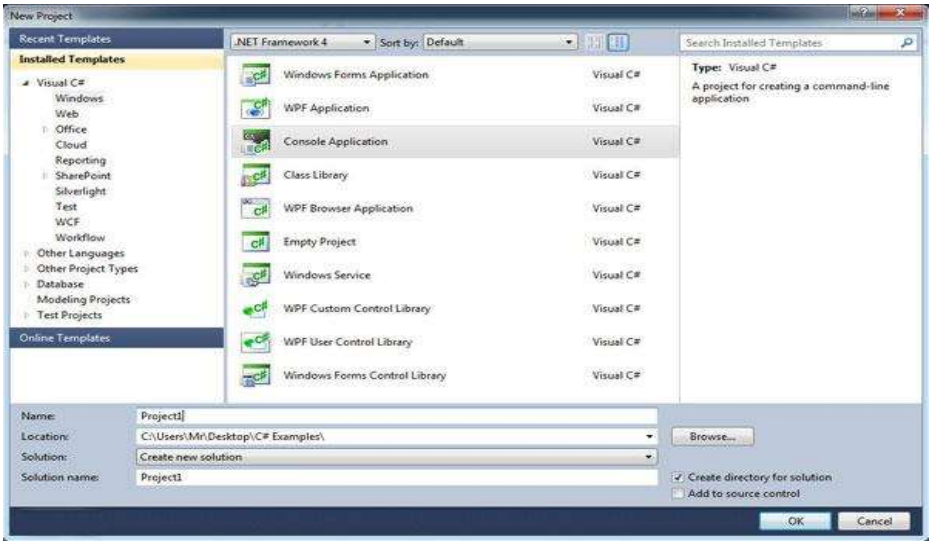
در این مقاله آموزشی، قصد مقایسه سی شارپ را با جاوا و دیگر زبان‌ها نداریم و تنها به آموزش این زبان می‌پردازیم به‌نحوی‌که برای شما مفید باشد. در این سری از مقالات آموزشی نیازی نیست از قبل با هیچ زبان برنامه‌نویسی آشنایی داشته باشید؛ چراکه از ابتدا و صفر شروع خواهیم کرد، اما در صورت آشنایی قبلی یادگیری برای شما آسان‌تر خواهد بود. لازم به ذکر است که در این مقالات، برنامه‌نویسی ما تحت محیط Console بوده و اگر شما مایل به یادگیری این زبان شیرین باشید و تمام قسمت‌های را دنبال کنید ما با مثال‌ها و تمرینات مختلف تا حد مقبولی به شما خوانندگان عزیز این زبان برنامه‌نویسی را آموزش می‌دهم.

قسمت اول

در این قسمت قصد داریم بیشتر با کدنویسی و محیط برنامه Visual Studio IDE آشنا شویم. ویژوال استودیو، IDE شرکت مایکروسافت است و IDE مخفف Integrated Development Environment (محیط یکپارچه توسعه نرم‌افزار) است.

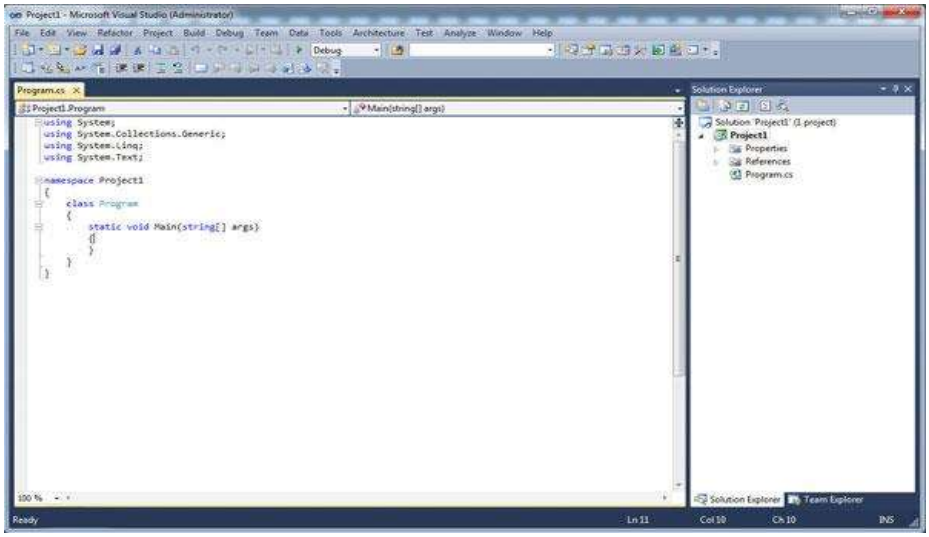
Visual Studioبه شما اجازه ویرایش، کامپایل، اجرا و خطایابی (Debug) برنامه‌های سی شارپ را می‌دهد پس برای دنبال‌کردن این سری مقالات و یادگیری C# نیاز دارید که یک نسخه از ویژوال استودیو را روی سیستم خود نصب کنید و پیشنهاد من نسخهٔ ۲۰۱۰ به بعد است؛ چراکه ممکن است در برخی از موارد از ویژگی‌های C# 4.0 استفاده کنیم. همچنین می‌توانید نسخهٔ رایگان را از وب‌سایت مایکروسافت دریافت کنید. دیگر وقت آن رسیده است که اولین برنامه سی شارپ را بنویسیم. برای این منظور باید یک پروژه برای سی شارپ در ویژوال استودیو بسازیم. پس مراحل زیر را برای ساخت یک پروژه خالی سی شارپ که در اینجا از Visual Studio 2010 Professional استفاده شده است دنبال کنید.

۱. ویژوال استودیو را اجرا کنید و از منوی بالا به File => New => Project بروید، سپس Windows را از لیست قالب‌های نصب شده انتخاب کنید، بعد از آن Console Application را انتخاب کنید:



سپس نام پروژه خودتان را در قسمت Name وارد کنید و روی OK کلیک کنید تا پروژه شما ساخته شود.

۲. زمانی که پروژه شما ساخته شد، ویژوال استودیو بدین شکل به نظر می‌رسد:



اگر به هر دلیلی پنجره Solution Explorer را در سمت راست مشاهده نکردید می‌توانید از منوی View با انتخاب Solution Explorer آن را فعال کنید.

تا اینجا شما توانستید پروژه خودتان را که هم‌اکنون آماده برای کدنویسی است تهیه کنید. همان‌طور که مشاهده می‌کنید مقداری کد به‌صورت پیش‌فرض در پنجره Program.cs قرار دارد که در ادامه به شرح مختصر آنها می‌پردازیم.

```
1                                                    using System;
```

این خط کد مشخص می‌کند که برنامه شما از فضای نام System استفاده می‌کند که یکی از فضاهای نام (namespaces)پایه‌ای در .Net Framework است. احتمالاً هنوز نمی‌دانید .Net Framework چیست، پس حتماً مقالهٔ چهار چوب دات نت را در ویکی‌پدیا مطالعه بفرمایید.

Namespace روشی برای مدیریت کد و گروه‌بندی کلاس‌های مرتبط به هم است.

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

توضیح این چند خط کد در مقالات بعدی و در جای خود دنبال خواهد شد.

```
class Program
```

این خط کد از کلمه کلیدی class برای اعلان یک کلاس جدید که از قبل در .Net Framework تعریف شده است استفاده می‌کند. Program نام این کلاس است. تعریف کلاس با آکولاد باز { شروع و با آکولاد بسته } تمام می‌شود که عناصر بین آکولاد، اعضای کلاس هستند. از آنجا که سی شارپ یک زبان برنامه‌نویسی

تماماً شیءگرا است و همهٔ سروکارش با کلاس‌ها است باید در نحوهٔ تعریف کلاس و استفاده از آنها تسلط کافی داشته باشیم که این کار را در مقالات آینده انجام خواهیم داد.

```
static void Main(string[] args)
```

متد Main() در اینجا قسمتی است که عملیات اصلی برنامه در آن انجام می‌شود. بدون متد Main() برنامه‌های سی شارپ قابل اجرا نخواهند بود، متد Main() هم با آکولاد باز و بسته محدوده خودش را مشخص می‌کند. در مورد Method و Class بعداً مفصل توضیح خواهیم داد پس اصلاً نگران نباشید، زیرا تا اینجا فقط یک توضیح مختصر در مورد کدهای پیش‌فرضی بود که در ابتدا می‌دیدید.

اکنون اندکی با برنامه Visual Studio IDE و نحوهٔ پروژه ساختن آشنا شدید. همین‌طور کدهای پیش فرضی که در ابتدا می‌دیدید را تاحدی شناختید. اکنون قصد داریم با چند کد ساده شروع به برنامه‌نویسی کنیم، ممکن است در ابتدا چیزی ببینید که اصلاً از آن سر درنمی‌آورید، اما اصلاً نترسید چون همه‌چیز را تا حدی که مفهوم مطلب بیان شود و اطمینان حاصل شود که شما خوانندگان عزیز آن موضوع را درک کرده‌اید، توضیح خواهیم داد. فقط از خواندن بازنایستید! همان‌طور که گفته شد عملیات اصلی برنامه در متد Main() نوشته می‌شود و فعلاً با بقیهٔ قسمت‌ها کاری نداریم؛ چراکه در جای خود به آنها می‌رسیم و به شرح آنها می‌پردازیم.

اکنون که با ساختن پروژه آشنا شدید، یک پروژه جدید بسازید.

به کد زیر نگاه کنید:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Project1
{
    class Program
    {
        // A C# program begins with a call to Main().
        static void Main(string[] args)
        {
            /*
                This is a simple C# program.
                Green lines are comments.
            */

            Console.WriteLine("A Simple C# Program.");
                                                                    }
                                                                    }
                                                                    }
```

همان‌طور که مشاهده می‌کنید مقداری از کدها به‌صورت پیش‌فرض در صفحه ما بوده‌اند و فقط این چند خط جدید است:

```
// A C# program begins with a call to Main().

/*
This is a simple C# program.
Green lines are comments.
*/
Console.WriteLine("A Simple C# Program.");
```

خطوط سبزرنگی که در بالا مشاهده می‌کنید comment (توضیح) هستند. در سی شارپ ما همانند سایر زبان‌های برنامه‌نویسی اجازه داریم که از کامنت استفاده کنیم. همان‌طور که می‌بینید به دو طریق می‌توانیم commentبگذاریم. یکی نوشتن جلوی دو اسلش // و دیگری نوشتن مابین ستاره و اسلش /* */ محتوای کامنت‌ها توسط کامپایلر نادیده گرفته می‌شوند، کامنت‌ها فقط برای این منظور توسط برنامه‌نویس نوشته می‌شوند که هرکس در حال خواندن کد و برنامه است آن قسمت از کد را بفهمد، زیرا کامنت‌ها اکثراً توضیح و شرحی در مورد کد هستند و نوشتن آنها اختیاری است.

```
Console.WriteLine("A Simple C# Program.");
```

خط کدی که در بالا مشاهده می‌کنید، رشتهٔ "A Simple C# Program." را در یک خط جدید در خروجی نمایش می‌دهد. فعلاً در همین حد بدانید که Console.WriteLine() در این مثال یک String (رشته‌ای از کاراکترها) را در خروجی نمایش می‌دهد. بعداً متوجه خواهید شد که چطور این اتفاق می‌افتد. پس دانستید که String رشته‌ای از کاراکترها است که حتماً باید بین این علامت (" ") بخوانید دابل کوتیشن، (Double quotation)قرار بگیرد و Console.WriteLine()برای چاپ کردن اطلاعات و پیغام در خروجی است و آنچه را که می‌خواهید در خروجی نمایش دهید باید بین () قرار دهید، درست طبق مثال. در انتهای آن نقطه و ویرگول بدین شکل; قرار می‌گیرد که اصطلاحاً به آن Semicolon (بخوانید سمی کالن) گفته می‌شود. بدین‌صورت:

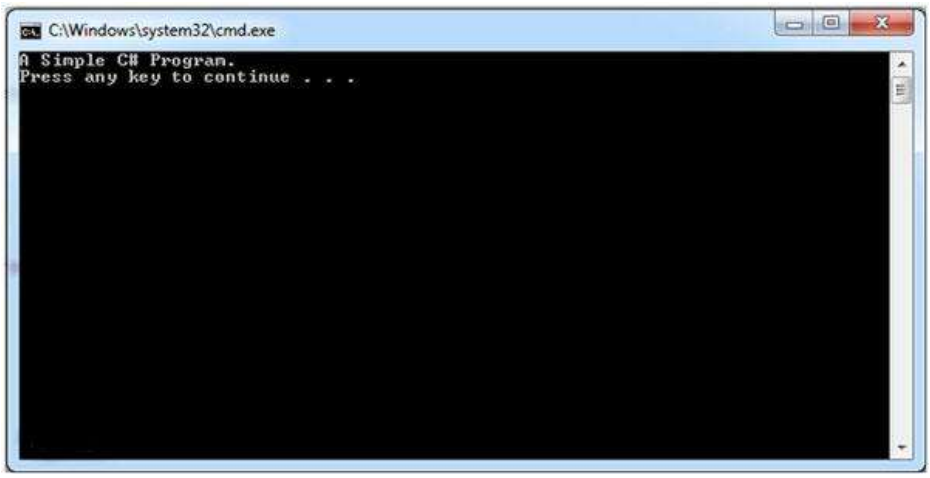
```
Console.WriteLine();
```

علامت semicolon در پایان هر دستور قرار می‌گیرد و هر دستور به‌صورت جداگانه در قسمت پایانی خودش از semicolonاستفاده می‌کند. در طول دوره آموزش با این روند آشنا خواهید شد که در چه جاهایی باید از semicolonاستفاده کنید.

توجه داشته باشید که سی شارپ به حروف بزرگ و کوچک حساس و اصطلاحاً case-sensitive است و این بدین معنی است که اگر فراموش کنید کجا باید از حروف بزرگ و کوچک استفاده کنید برنامه شما دچار خطا می‌شود. برای مثال اگر نام یک متغیر را phoneNum انتخاب کرده باشید و در جای دیگر برای استفاده از آن بنویسید PhoneNum یا Phonenum آنگاه برنامه شما دچار مشکل می‌شود. دستور Console.WriteLine() هم به همین روال است همین‌طور بقیه دستورها، هرچند که کامپایلر سی شارپ فوق‌العاده قدرتمند است و در صورت بروز کوچکترین خطا شما را با خبر می‌کند، به جز خطاهای منطقی. در مورد خطاهای منطقی بعداً بیشتر صحبت خواهیم کرد. تا اینجا با یک سری از اصطلاحات و مفاهیم آشنا شدید، بعدازاینکه یک سری از مطالب را با هم مورد بررسی قرار دادیم سراغ یک مثال درست‌وحسابی می‌رویم.

```
static void Main(string[] args)
{
    Console.WriteLine("A Simple C# Program.");
}
```

پس طبق کد بالا دستور چاپ یک string را بنویسید و سپس Ctrl + F5 را با هم بگیرید تا خروجی برای شما نمایش داده شود) :به‌جای فشردنCtrl + F5 ، از منوی Debug با انتخاب Start Without Debugging هم می‌توانید این کار را انجام دهید.



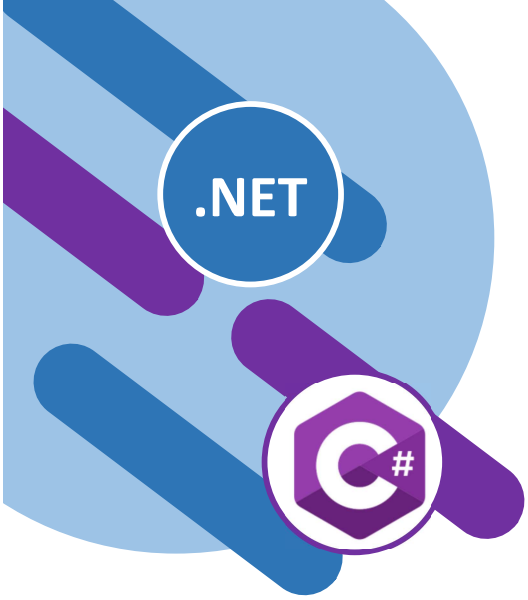
همان‌طور که می‌بینید A Simple C# Program در پنجره خروجی نمایش داده شد.

هدف از این سری مقالات آموزشی یادگیری سی شارپ به‌طور گام‌به‌گام و تقریباً کامل برای هر مبحث است. هرچند مجبوریم بعضی از مسائل را فعلاً به‌صورت کامل باز نکنیم تا در فهم موضوع دچار مشکل نشوید، اما در آینده به آنها پی خواهید برد. اگر این سری مقالات را همیشه دنبال کنید کم‌کم هر قسمت از زبان سی شارپ را به‌طور مفهومی آموزش می‌بینید. اگر در مورد مطلب هر قسمت سؤالی برایتان پیش آمد و مشکلی داشتید

می‌توانید در قسمت نظرات سؤالات خود را مطرح کنید یا از شبکه‌های اجتماعی که در پروفایل من در همین سایت موجود است با من تماس بگیرید.

در قسمت بعدی با value type، متغیرها و عملگرها آشنا می‌شویم.

به یاد داشته باشید آنکه می‌خواهد روزی پریدن آموزد، نخست می‌باید ایستادن، راه رفتن، دویدن و بالا رفتن آموزد. پرواز را با پرواز آغاز نمی‌کنند.



قسمت دوم

در قسمت قبل با یک برنامهٔ ساده شروع کردیم که تنها در خروجی یک پیغام را چاپ می‌کرد، در این قسمت قصد داریم با متغیرها(variables) ، عملگرها (operator) و value type بیشتر آشنا شویم.

شاید هیچ سازه‌ای به اندازهٔ متغیرها برای یک زبان برنامه‌نویسی اهمیت نداشته باشند. متغیر درواقع مکانی در حافظه است که می‌توان یک مقدار را به آن اختصاص داد. به این دلیل به آن متغیر می‌گویند که مقدار آن می‌تواند در طول اجرای برنامه تغییر کند. به‌عبارت‌دیگر محتوای متغیرها قابل تغییر هستند و همیشه ثابت نمی‌مانند.

ویژوال استودیو را اجرا کنید و یک پروژه جدید بسازید اگر این کار را نمی‌توانید انجام دهید حتماً قسمت قبل را مطالعه بفرمایید.

به برنامه زیر دقت کنید، این برنامه دو متغیر را که اسم آنها x و y است می‌سازد:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Example
{
    class Program
    {
        static void Main(string[] args)
        {
            int x;
            int y;

            x = 100; // this assign 100 to x

            Console.WriteLine("X Contains " + x);
```

```
            y = x / 2;

            Console.Write("y contains x / 2: ");
            Console.WriteLine(y);
        }
    }
}
```

وقتی برنامه را اجرا کردید (توسط Ctrl + F5) خروجی زیر برای شما نمایش داده می‌شود:

```
x contains 100
y contains x / 2: 50
```

این برنامه شامل چندین مفهوم جدید است که در ادامه به شرح آنها می‌پردازیم:

```
int x;
```

اعلام یک متغیر که نام آن x و نوع آن عدد صحیح (integer) است. در سی شارپ، همهٔ متغیرها پیش از اینکه مورد استفاده قرار بگیرند ابتدا باید اعلام شوند. علاوه‌بر این، نوع مقداری را که متغیر در خود نگه می‌دارد باید مشخص شود. در این مورد x می‌تواند مقادیر integer (عدد صحیح) را در خود نگه دارد. در سی شارپ، برای اعلام یک متغیر از نوع عدد صحیح (integer) قبل از نام آن از کلمه کلیدی int استفاده می‌کنند. پس int x یک متغیر را اعلام می‌کند که اسم آن x و نوع آن integer (عدد صحیح) است.

```
int y;
```

توجه داشته باشید که در این مورد هم یک متغیر از نوع عدد صحیح ساخته می‌شود با این تفاوت که اسم آن فرق می‌کند. به‌صورت کلی شما برای اعلام یک متغیر از این الگو می‌توانید پیروی کنید:

```
type var-name;
```

کد type نوع متغیر و var-name نام متغیر را مشخص می‌کند. علاوه‌بر int سی شارپ از نوع‌های داده‌ای (data type)دیگر هم پشتیبانی می‌کند که کامل به شرح آنها خواهیم پرداخت.

```
x = 100;
```

این خط کد مقدار ۱۰۰ را به x اختصاص می‌دهد. در سی شارپ، عملگر انتساب یک تک مساوی است و بدین‌صورت عمل می‌کند که یک کپی از مقدار سمت راست خودش را در متغیر سمت چپش قرار می‌دهد. همان‌طور که می‌بینید در اینجا علامت انتساب باعث شده است که یک کپی از مقدار سمت راست (عدد ۱۰۰) در متغیر سمت چپ متغیر x قرار بگیرد.

خط کد بعدی مقدار x را در خروجی نمایش می‌دهد که قبل از آن رشتهٔ "X Contains " قرار دارد:

```
Console.WriteLine("x contains " + x);
```

در این خط کد علامت + موجب می‌شود که مقدار x بعد از string (رشته) نمایش داده شود. این روش می‌تواند یک حالت کلی به خود بگیرد. با علامت + شما می‌توانید آیتم‌های دیگری را که می‌خواهید در یک WriteLine()داشته باشید، به هم وصل کنید.

خط کد بعدی مقدار x را که بر ۲ تقسیم شده است به y اختصاص می‌دهد:

```
y = x / 2;
```

این خط کد مقدار x را بر ۲ تقسیم و سپس نتیجهٔ آن را در y ذخیره می‌کند؛ بنابراین بعد از اجرای این خط، y شامل مقدار ۵۰ می‌شود و مقدار x بدون تغییر باقی می‌ماند. سی شارپ مانند تمام زبان‌های برنامه‌نویسی طیف گسترده‌ای از عملگرهای محاسباتی را پشتیبانی می‌کند. ازجمله این عملگرها:

+	Addition
-	Subtraction
*	Multiplication
/	Division

دو خط بعدی برنامه:

```
Console.Write("y contains x / 2: ");
Console.WriteLine(y);
```

دو چیز در اینجا جدید است: اول Console.Write() است که رشتهٔ y contains x / 2را در خروجی نمایش می‌دهد. تفاوت این دستور با دستور Console.WriteLine() این است که وقتی خروجی بعدی تولید می‌شود از همان خط شروع می‌شود، نه از خط بعدی. دوم اینکه Console.WriteLine() متغیر y را به‌عنوان ورودی گرفته است و مقدار آن را در خروجی نمایش می‌دهد. پس توجه داشته باشید، هنگامی‌که می‌خواهید مقدار یک متغیر را در خروجی نمایش دهید کافی است اسم آن را در Console.WriteLine() یا Console.Write() وارد کنید. همان‌طور که مشاهده می‌کنید مقدار y در جلوی رشتهٔ y contains x / 2نمایش داده می‌شود زیرا برای چاپ این رشته از Console.Write() استفاده کردیم، ولی اگر از Console.WriteLine() استفاده می‌کردیم مقدار y در خط بعدی نمایش داده می‌شد.

یک نکتهٔ جالب دیگر در مورد اعلام متغیرها این است که شما می‌توانید دو یا بیشتر از دو متغیر را در یک تعریف متغیر اعلام کنید. فقط کافی است اسم متغیرها را با کاما از هم جدا کنید، برای مثال متغیرهای y و x را از این روش هم می‌توان اعلام کرد:

```
int x, y;
```

همچنین نیازی نیست که ابتدا متغیر را تعریف کنید و بعد به آن مقدار بدهید، می‌توانید این کار را هم زمان با تعریف متغیر انجام دهید. بدین‌صورت:

```
int x = 100;
```

تا اینجا ما فقط از اعداد صحیح برای محاسبه استفاده کردیم، مثل عدد ۱۸ یا ۲۲ و غیره. برای استفاده از اعداد اعشاری مثل ۲۲٫۵ یا ۱۵٫۶ باید از نوع عددیdouble ، floatو decimal استفاده کنیم. نوع عددی float برای مقادیر ممیز شناور با دقت کمتر و نوع عددی double برای مقادیر ممیز شناور با دقت بیشتر است به‌طوری‌که دقت double دو برابر float است decimal .یک دیتا تایپ ۱۲۸ بیتی است که در مقایسه با floatو double از دقت بیشتر و بازهٔ کمتری برخوردار است و این خاصیت آن باعث می‌شود که برای محاسبات پولی و مالی بسیار مناسب باشد.

نوع float سی و دو بیتی است و بازهٔ اعداد آن را می‌توانید در جدول زیر مشاهده کنید:

Type	Approximate range	Precision
float	$\pm 1.5 \times 10^{-45}$ to $\pm 3.4 \times 10^{38}$	7 digits

نوع دابل ۶۴ بیتی است و محدودهٔ آن در جدول زیر مشخص است:

Type	Approximate range	Precision
double	$\pm 5.0 \times 10^{-324}$ to $\pm 1.7 \times 10^{308}$	15-16 digits

بازهٔ تقریبی و دقت decimal را در جدول زیر می‌توانید مشاهده کنید:

Type	Approximate Range	Precision
decimal	$\pm 1.0 \times 10^{-28}$ to $\pm 7.9 \times 10^{28}$	28-29 significant digits

سی شارپ شامل دو دسته‌بندی برای data type (نوع داده) است:

- Value Type
- Reference Type

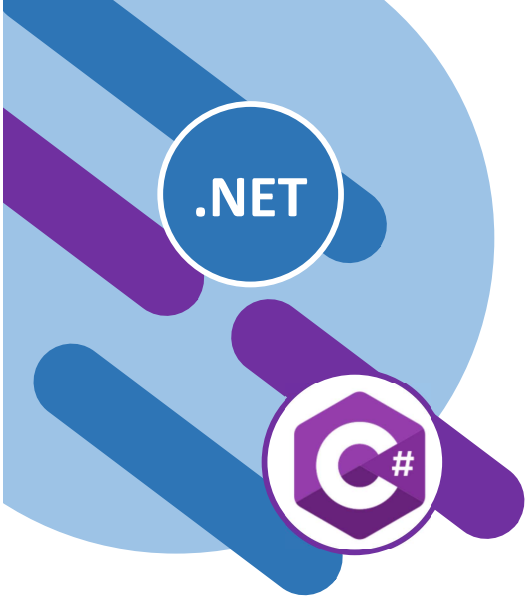
تفاوت بین این دو، مقداری است که آن متغیر دارد. برایvalue type ، متغیر یک مقدار واقعی را در خود نگه می‌دارد مثل: ۲۲ یا ۳۴۵٫۶ اما برایReference type ، متغیر یک ارجاع به Value را در خود نگه می‌دارد که این اکثراً در کلاس مورد استفاده قرار می‌گیرد. در مورد Reference type بعداً بیشتر صحبت خواهیم کرد، فعلاً به value type می‌پردازیم.

در جدول زیر لیست کاملی از تمام value type ها را می‌بینید:

Type	Description	Range
Byte	8-bit unsigned integer	0 to 255
Sbyte	8-bit signed integer	-128 to 127
Short	16-bit signed integer	-32,768 to 32,767
Ushort	16-bit unsigned integer	0 to 65,535
Int	32-bit signed integer	-2,147,483,648 to 2,147,483,647
Uint	32-bit unsigned integer	0 to 4,294,967,295
long	64-bit signed integer	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
Ulong	64-bit unsigned integer	0 to 18,446,744,073,709,551,615
float	32-bit Single-precision floating point type	-3.402823e38 to 3.402823e38
double	64-bit double-precision floating point type	-1.79769313486232e308 to 1.79769313486232e308
decimal	128-bit decimal type for financial and monetary calculations	(+ or -)1.0 x 10e-28 to 7.9 x 10e28
char	16-bit single Unicode character	Any valid character, e.g. a, *, \x0058 (hex), or\u0058 (Unicode)
Bool	8-bit logical true/false value	True or False
Object	Base type of all other types.	
string	A sequence of Unicode characters	

سی شارپ همچنین ۹ integer type را تعریف می‌کند که char ، byte، sbyte، short، ushort، int، uint، long، ulongهستند و هرکدام بازهٔ خاصی از اعداد را شامل می‌شوند. در جدول زیر بازهٔ عددی و تعداد بیت هریک را می‌بینید. نوع char اصولاً به نمایندگی از کاراکتر استفاده می‌شود که در قسمت‌های آینده در مورد آن صحبت خواهیم کرد:

در قسمت بعدی با چندین مثال به تشریح کامل‌تر مباحثی که در اینجا به‌صورت تئوری بیان شد می‌پردازیم. همچنین فراموش نکنید که حتماً سؤالات و مشکلات خود را در هر قسمت بیان کنید تا موضوع برای شما و دیگر خوانندگان عزیز قابل‌فهم‌تر شود. ضمناً، سعی کنید که حتماً اصطلاحات انگلیسی استفاده شده در این مقالات را یاد بگیرید؛ چراکه برنامه‌نویسی اصطلاحات زیادی دارد و اکثراً واژهٔ معادل فارسی ندارند یا در صورت داشتن، معادل فارسی آنها آن‌چنان‌که باید مناسب نیست و ما ترجیح می‌دهیم که از واژه و اصطلاحات اصلی آنها استفاده کنیم.



قسمت سوم

در قسمت قبل توضیحات مختصری در مورد value type، متغیر (variable) و عملگرها (operators) داده شد. در این قسمت با چند مثال به تشریح کامل‌تر آنها خواهیم پرداخت. همان‌طور که در قسمت قبل گفته شد، data types به دو دستهٔ value types و reference types تقسیم می‌شوند و دانستید که سیزده value types داریم.

درمجموع به این سیزده ولیو تایپ، simple types می‌گویند و دلیل این نامگذاری این است که اینها شامل مقدار تکی (singleValue) هستند و به‌عبارت‌دیگر، ترکیبی از دو یا بیشتر از دو مقدار نیستند.

Float و Double

به مثال زیر توجه کنید:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ExFloat
{
 class Program
 {
 static void Main(string[] args)
 {
 int i = 25;
 float f = 16.8F;
 float result = i / f;
 Console.WriteLine("Result is: " + result);
 }
 }
}
```